CONTROL OF C2 UNIT USING ARENA MODELING AND SIMULATION

Sudha Thavamani

Dept. of Electrical and Computer Engineering
Binghamton University
Binghamton, NY 13902, U.S.A.

ABSTRACT

This paper reports the results of a simulation study using Arena on a system resembling a part of the high level supporting structure of a Command and Control Center for military air operations. The system is modeled as a reconfigurable queuing network with servers that are subject to failures and with a human operator unit. The main interest is to understand and quantify the benefits of reconfiguration, and to recommend changes in structural parameters and policies for optimized network performance. An optimal control policy is applied to the main processing unit in C2. The investigation is based on a Markov decision problem with the C2 unit as a closed queuing network; where in the structure of optimal policy is accomplished by means of dynamic programming. The paper presents our modeling effort, and simulation results that compare the C2 system performance.

1 INTRODUCTION

A Command and Control (C2) Center plays the role in an air operation as the controller in a feedback system. It carries out the functional mapping from information to decision in the feedback loop. A great deal of effort has been directed toward the design of the controller to maximize the probability of success (Cruz et al. 2001, Wohletz et al. 2001). The underlying assumption has been that the structure supporting the functional mapping in C2 is always intact. In reality, however, a typical C2 center has grown to be a large and complex system, and this system can experience many types of failures. Analysis by Wu et al. (2004a) showed that unavailability in the C2 supporting structure could greatly impact the outcome of an air operation, and that reconfiguration is an effective way to enhance the availability. The study, however, is confined to very simple systems under most simplified assumptions. The system under study in this paper, which is modeled as a closed queuing network (Cassandras et al. 1999) shown in Figure 1, resembles a high level supporting structure in C2. It consists of a database unit containing two non-overlapping da-tabase resources named database1 and database2, a human operator unit, a reconfigurable processing unit containing two processing cells in series. The human operator unit in the network is modeled as a simple queuing node with queue length dependent service rate and rework probability. The nonhuman servers in the database unit and the processing unit are all subject to failures. In processor-failure scenarios, reconfiguration was shown to lessen performance degradation.

In comparison with a more detailed analytic study on the processing unit (Wu et al. 2004b), the use of a simulation tool, in this case Arena (2004), allows us to study a larger network, and more general types of distributions for service time, server time to failure and to restoration. The performance analysis for large wireless network using Arena software was performed for different application in Wu et al. (2006). The goal of this paper is to understand and quantify the benefits of reconfiguration, and to recommend changes in network architecture, structural parameters, and operating policies for better network performance. In particular, the expected steady state response time and availability are considered in our study as the network performance measures.

Another portion of the paper deals with developing the theory of optimal control for which we limit ourselves to the main processing unit of the C2 system, illustrated in Figure 6. The simulation results are shown for the processing unit where in the structure of the optimal control policy is implemented. The optimal policy is a *threshold-based* policy that is demonstrated to minimize the average customer waiting time. The threshold based policy establishes a control wherein a particular processing cell is chosen based on the cost when it reaches or exceeds a certain threshold. The threshold based optimality, to refer a few in the literature, has been exposed for a single server queuing system (Deb et al. 1973) and multiple server system (Pepyne et al. 1997). In this paper, we consider a multiple server, multi- class, closed queuing system analysis with each server having a sufficient storage capacity. The simulation analysis is based on modeling the processing unit in Arena (2004) as a Markov decision problem (MDP) (Cas-

sandras et al. 1999) and applying dynamic programming techniques (Cassandras et al. 1999) to derive the structural properties of the optimal control policy. The objective is to demonstrate the significance of optimal control to maximize the system performance. The paper exhibits the superior performance of the C2 processing unit as desired due to the optimal control policy with respect to response time of the customers, waiting time of the customers in the queue, number of customers waiting in the queue and the server utilization.

The paper is organized as follows. Section 2 describes the model building of high level supporting structure of C2 in Arena and the parameters and operating policies applied to the network in Figure 1. The section presents the reconfigurable control results of output analysis from Arena simulations. Section 3 discusses the performance based on optimal control policy applied to the main processing unit of C2. The simulation model, experiments, and observations are presented and discussed in this section. The paper terminates following the conclusions in section 4. Appendix A provides the C2 processing unit parameters used in model building, Appendix B presents the one step cost and Appendix C offers the optimality equations that are taken advantage in the simulation study.

2 RECONFIGURABILITY OF HIGH LEVEL C2 UNIT

The modeling in Arena (2004) of the C2 system and the simulation output analysis is discussed in the section. The circles in Figure 1 marked by DB1, DB2, PC1, PC2 are servers. They are specified by service time distributions, as well as up and down time distributions. The human operator HO is a special server specified by a state dependent service time distribution and a state dependent rework probability in case of unsatisfactory jobs. The groups of delays are placeholders for returning customers to prepare for the next round of services. They are specified by delay time distributions. The queues are represented by open boxes in Figure 1, and are of unlimited capacity. Q_{DB1}, Q_{DB2}, Q_{HO}, Q_{PC1}, Q_{PC2} are the queue lengths of the nodes they belong, including the customers being served, and the customers in the queues are served on a first-come-first-served basis. The model in Figure 1 is built with Arena (2004), which offers a wide selection of process and data modules, distributions for service, delay, failure, and restoration times, as well as output statistics.

Arena simulations were run for wide ranges of parameters. All rates are associated with the exponential distribution with μ, λ, γ, ν denoting service, delay, restoration, and failure rates, respectively. Superscript i indicate the association of a rate with an intermittent failure or restoration process. Table 1 below gives the set of parameters and distributions used to produce the reported outcomes in the Figure 1 and Figure 2. The routing probability is $p=0.5$ in

the database unit. The rework probability in the human operator unit is:
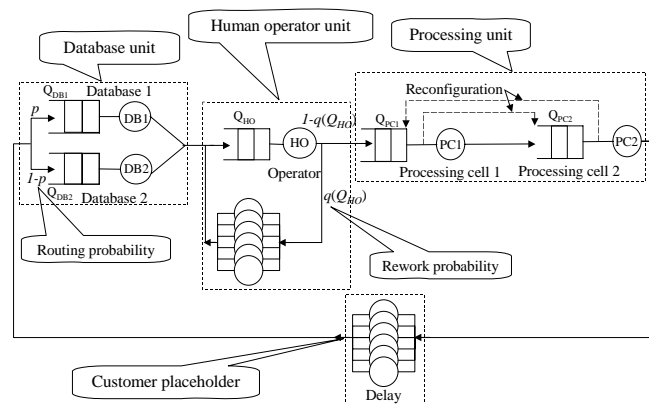
$$q(Q_{HO}) = .99\exp(-.05Q_{HO}).$$

Figure 1: A Queuing Network Model of Interconnected C2 Units

Table 1: Input Probability Distributions to Arena Simulation of the Network

	μ (min^{-1})	ν (hr^{-1})	γ (hr^{-1})	ν^i (hr^{-1})	γ^i (hr^{-1})
DB1,2	1/2.8	1/500	1/24		
PC1,2	1/16	1/500	1/24	1/100	¼
Delay	$\lambda=1/5$				
HO	.9(1-.9exp(-.9Q_{HO}))				

Table 2 describes the operating policy implemented in the processing unit. The policy states, in simple terms, that whenever a failure in a processing cell occurs, the remaining server serves customers in Q_{PC2} first, and then resume the service for the head of the queue customer in Q_{PC1} as soon as Q_{PC2} is emptied.

Table 2: Operating Policy in the Processing Unit When a Server Fails

PC1 State	PC2 State	Reconfiguration
Intact, empty	Failed, queued	PC1← Q_{PC2}
Intact, queued	Failed, queued	PC1← Q_{PC2}, Q_{PC1}
Failed, queued	Intact, empty	PC2← Q_{PC1}

Output statistics are collected and estimates are obtained for the expected steady state response time and the expected steady state availability of the network with and without reconfiguration.
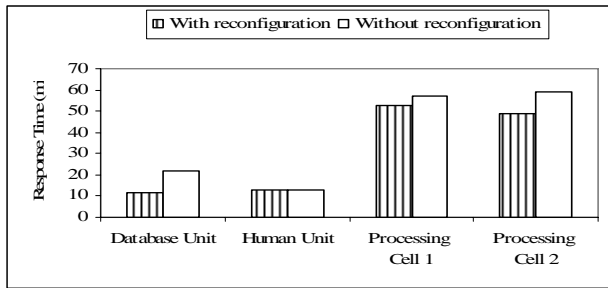
Figure 2: Response Time Comparison with & without Reconfiguration with Exponential Rates
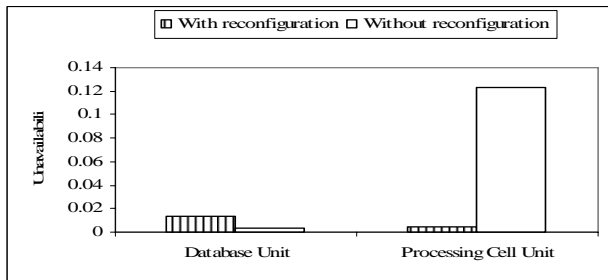

Figure 3: Availability Comparison with and without Reconfiguration with Exponential Rates

The response time of the network is defined as the amount of time elapsing from the instant a customer enters the queue in the database unit until it completes the service (Cassandras et al. 1999). There is substantial decrease, as seen in Figure 2, in response time of the system when reconfiguration is applied. The availability is the fraction of time when all units in the network are up that have customers in waiting. A significant increase in availability is achieved when the system is reconfigured as seen from Figure 3.

An alternative set of distributions chosen can be argued (Kelton et al. 2004, Law et al. 2000) to be more appropriate for a C2 system. For instance, triangular distributions for the service time of HO and for delay unit; Gamma distributions for the service times of DB1, DB2, PC1, and PC2, and for the restorations times of PC1 and PC2; and Weibull distributions for failure times of DB1, DB2, PC1, and PC2. Table 3 presents the parameters used for the purpose of simulation with these distributions. For example, the most commonly used distribution for a unit life is the exponential distribution. An extension from exponential to Weibull distribution allows a more truthful description of a unit life in our application where an unit that is found to be good after some usage will have a shorter residual life than a brand new unit, while the used unit found to be good is indistinguishable from a new one if it is modeled by an exponential life time distribution. For the performance measures of interest, consistent results have been observed.

Table 3: Input Probability Distributions to Arena Simulation of the Network

	$1/\mu$ (min)	$1/\nu$ (hr)	$1/\gamma$ (hr)	$1/\nu^i$ (hr)	$1/\gamma^i$ (hr)
DB1,2	Gamma (2.5454,1.1)	Gamma (21.8181, 1.1)	Weibull (500, 1.1)		
PC1,2	Gamma (14.5454,1.1)	Gamma (21.8181, 1.1)	Weibull (500, 1.1)	Gamma (3.6363, 1.1)	Weibull (100, 1.1)
Delay	Triangular (0.5,5,10)				
HO	$1/(.9(1-.9\exp(-.9Q_{HO})))$				

These collection of non-exponential distributions employed is determined to be more appropriate for the C2 system, further dropping the expected steady state unavailability as seen in Figure 4. The expected steady state response time of the network with reconfiguration is lower than without the reconfiguration, as displayed in Figure 5. The availability improvement is more evident than the response time improvement.
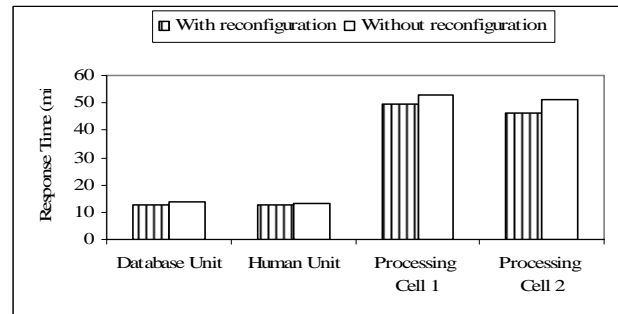

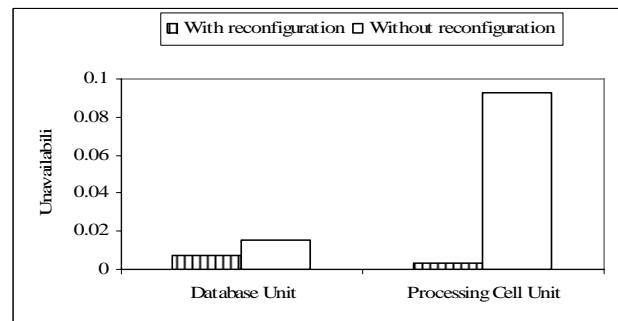Figure 4: Response Time Comparison with & without Reconfiguration

Figure 5: Unavailability Comparison with and without Reconfiguration

3 OPTIMAL CONTROL OF C2 PROCESSING UNIT

The work presented in the section is a continuation of earlier effort in Wu et al. (2004b), wherein a supervisory control was applied to the C2 processing unit. The C2 processing unit as a closed queuing network, as shown in Figure 6, is considered.

3.1 Modeling of C2 Processing Unit

The C2 processing unit contains two cells in series, performing two different tasks in sequence. There are total of 3 customers in the network and arriving customers, classified as class I, are served according to FCFS basis. The customer completing the task at cell 1 is promoted as class II customer and the elevated status of this customer is eliminated after completing the task at cell 2. Each cell consists of a queue and a server. The server at cell i, named as Si, has an exponential service rate of μ_i, an exponential failure rate of ν_i and an exponential restoration rate of γ_i. The queue heading each server has a sufficient storage capacity. In the feedback branch, there are three servers that portray the delay aiming to signal the response times of the ignored nodes in the simplified model. Each delay unit is represented by an exponential rate of λ.

The main focus of this part of paper is to develop an optimal control policy for C2 system. The optimal policy developed does not depend on the arrival or service distributions of the model. The major concentration is to exhibit the structure of the optimal control policy which is a threshold - based policy that minimizes the discounted or average waiting time of the customer. The controller directs a customer to a particular server based on a threshold. The queuing problem is modeled based on Markov decision problem (Cassandras et al. 1999) and applying dynamic programming technique to derive the structural properties of the optimal control policy (Cassandras et al. 1999).
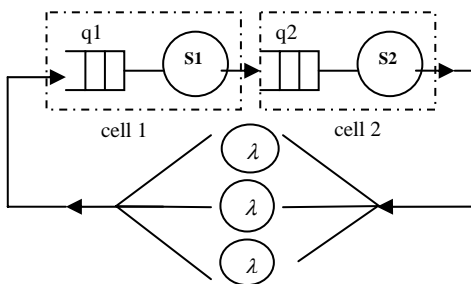

Figure 6: A C2 Processing Unit as a Closed Queuing Network

3.2 Markov Decision Problem and Dynamic Programming

The C2 processing unit has a state space $\chi = \{q_1 q_2 S_1 S_2\}$, where q_i is the queue length of server S_i, where $q_i \in \{0,1,2,3\}$ is the queue length at cell i with $q_1 + q_2 \leq 3$, and $S_i \in \{0,1\}$ represents server i which is either good ($S_i = 0$) or failed ($S_i = 1$). For this system *40* states [10] can be obtained during the occurrence of an event. Let $U(q_1 q_2 S_1 S_2)$ denote the set of admissible actions from the state $\chi = \{q_1, q_2, S_1, S_2\}$ when such events transpire, then the control action is specified as

$$U(q_1 q_2 S_1 S_2) = \begin{cases} u = 0, \text{ no action} \\ u = 1, \text{ customer from } q_2 \text{ switched to } q_1 \\ u = 2, \text{ customer from } q_1 \text{ switched to } q_2 \end{cases}$$

The uniformization technique (Cassandras et al. 1999) is used to convert the continuous time Markov decision problem (MDP) into an equivalent discrete-time MDP. This is achieved by choosing a uniform rate $\Gamma = \lambda + \sum_{i=1}^{2} \mu_i + \sum_{i=1}^{2} \nu_i + \sum_{i=1}^{2} \gamma_i$, which is the total event rate in the C2 processing unit as a closed queuing network. Let P_{ij} denote the transition probability from state i to state j and $C(q_1 q_2 S_1 S_2, u_k)$ denote the cost at k^{th} time step, when the control action taken at the beginning of the current time step is $u \in U(i)$. The Appendix B displays the one-step cost function, which is a function of $\{b_i, w_i, sw_i\}$, where b_i, w_i, sw_i are positive and bounded holding cost, wear & tear cost, switching cost respectively of server S_i. To minimize the total expected discounted cost, an optimal stationary policy π^* is attained

$$V_{\pi^*}^{\alpha}(i) = E_{\pi}\left[\sum_{k=0}^{\infty} \alpha^k C(q_1 q_2 S_1 S_2, u_k)\right], \quad (1)$$

where E[.] is the expectation operator, α *(0<α<1)* is a discount factor and i is the initial state. A policy that gives the least cost in equation (1) such that the optimality equations are satisfied is determined.

From the dynamic programming algorithm (Cassandras et al. 1999), the optimality equation is

$$V_{\pi^*}^{\alpha}(i) = \min_{u \in U(i)}\left[C(i,u) + \alpha \sum_{j}^{\infty} P_{ij}(u) V_{\pi^*}^{\alpha}(j)\right] \quad (2)$$

Under the same conditions there is an optimal stationary policy:

$$u^*(i) = \arg \min_{u \in U(i)} \left[C(i, u) + \alpha \sum_j^\infty P_{ij}(u) V_{\pi^*}^\alpha(j) \right] \quad (3)$$

The optimality equations for the each of possible *40* states in C2 processing unit are shown in Appendix C.

3.3 Simulation Analysis

The threshold policy is implemented using simulation and the performance of the optimality policy is evaluated. The simulation model is constructed wherein the arriving customer in the processing unit is directed to the respective cell depending on the minimum cost. For the system parameters specified in Appendix A, a plot of work in progress is sketched using output analyzer. From the plot, the run length of 5 days is determined to be sufficient for the model to reach the steady state. A warm-up period of 2 days is good to resolve the effect of the empty- and-idle initial conditions. The choice of warm up period can be proved using Welsh method (Law et al. 2000). The least cost policy is specified by the optimality equation based on the state of the queues and servers, as shown in the Appendix C. For example, at any point in the simulation, the state of the system is 3010, that is, there are 3 customers in q_1 and failed server S_1 with server S_2 empty and idle, a decision is taken based on the optimality equation, which is

V3010=min((c(3010,u=0)+(α γ_1 V3000)+(α v_2 V3011)+(α ($\lambda + \mu_1 + \mu_2 + v_1 + \gamma_2$)V3010)),(c(3010,u=2)+(α μ_1 V2010)+(α ($\lambda + \mu_2 + v_1 + v_2 + \gamma_1 + \gamma_2$)V3010)))   (4)

The minimum of the cost between the two terms in the bracket of equation (4) is chosen. If, for instance, the second term in the bracket is minimum, then the customers from q_1 are forwarded to processing cell *2*. The one step cost for this case would be *(2b1)+sw2+w2*, where *w2* is the cost of processing the customer of server S_1 in S_2; *sw2* is the switching cost incurred by S_2 and *b1* is the waiting cost of each of the customer.

The processing unit performance are measured in terms of response time of the customers, waiting time of the customers in the queue, number of customers waiting in the queue and the server utilization. It is evident from Figure 7 that the response time of the processing unit is least when the optimal policy is applied compared to the case with or without reconfiguration.

The simulation based on threshold based policy produces excellent outcomes in terms of waiting time of customers in queue (Figure 8) and number of customers waiting in the queue (Figure 9), in contrast to the case with and without reconfiguration. The advantage of opting for optimal control policy is also portrayed with regard to server utilization. The utilization of the servers is highest in the case with optimal control, as shown in Figure 10.
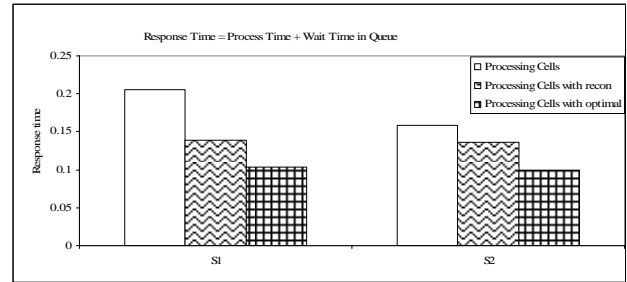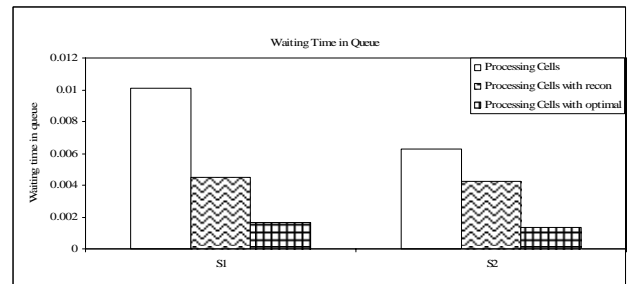

Figure 7: Response Time in a C2 Processing Unit

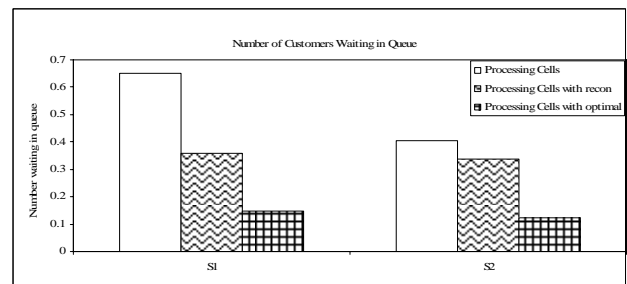
Figure 8: Wait Time in Queue in a C2 Processing Unit

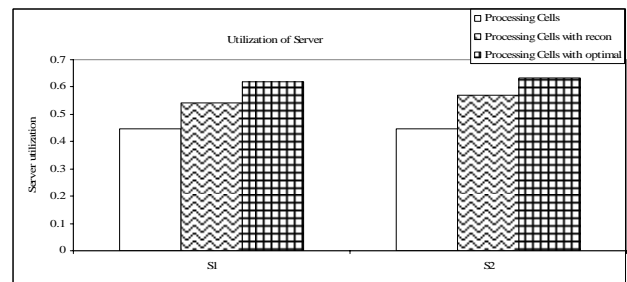Figure 9: Number waiting in queue in a C2 processing unit

Figure10: Utilization of a server in a C2 processing unit

4 CONCLUSION

In this paper, a system resembling a section of the high level supporting structure of a Command and Control (C2)

Center for air operations is analyzed and studied. The system is modeled in Arena as a queuing network with servers that are subject to failures and with a human operator unit. The simulation investigation reveals significant improvement in reconfigured system performance related to availability and response time. The alternative set of non-exponential distributions used is found to be more appropriate for the C2 system, further increasing the expected steady state availability and reducing the expected steady state response time of the network with reconfiguration. The availability improvement is more evident than the response time improvement.

The paper also discusses optimal performance of a main processing unit of the C2 system that was epitomized as a closed queuing system. Dynamic programming was employed to illustrate the optimal policy minimizing the discounted or average customer waiting time as a threshold based policy. From the simulation study, the advantage of optimal policy on the processing unit is examined and the results are found to be very encouraging. The optimal performance of the processing unit is compared against the case with and without the reconfiguration of the processing unit. The statistics show the optimal performance guarantees the best performance in C2 processing unit.

ACKNOWLEDGMENTS

APPENDIX A: C2 PROCESSING UNIT SIMULATION PARAMETERS

μ_1 =12, μ_2 =12, α =1, γ_1 =1/24, v_2 =0.005, v_1 =0.005, λ =6, γ_2 =1/24, sw1=2.0, sw2=5.0, w1=5.0, w2 =5.0, b1=5.0, b2=5.0

APPENDIX B: ONE-STEP COST OF C2 PROCESSING UNIT

c0000=0; c0001=0; c0010=0; c0100=w2; c1000=w1; c0110=w2; c0200=b2+w2; c1001=w1; c1100=w1+w2; c0011=0;c2000=b1+w1;c0111=b2;c0210=b2+w2; c0300=(2b2)+w2;c1011=b1;c1210=b1+b2+w2; c2001=b1+w1; c3000=(2b1)+w1;c0211=(2b2); c1110=b1+w2;c2100=b1+w1+w2;c1111=b1+b2; c2011=2b1;c2110=(2b1)+w2;c3001=(2b1)+w1; c1211=b1+(2b2);c2111=(2b1)+b2;c3011=3b1; c1200=w1+b2+w2;c0310=(2b2)+w2; c031=3b2;

c(0101, u=0)=b2; c(1010,u=0)=b1;c(0201, u=0)=(2b2); c(1101,u=0)=w1+b2;c(2010,u=0)=2b1;

c(0301,u=0)=3b2;c(1201,u=0)=w1+(2b2); c(2101,u=0)=b1+w1+b2;c(3010,u=0)=3b1; c(0101,u=1)=sw1+w1;c(1010,u=2)=sw2+w2; c(0201,u=1)=b2+(sw1)+(w1);c(1101,u=1)=b1+sw1+w1; c(2010,u=2)=b1+(sw2)+w2; c(0301,u=1)=(2b2)+w1+sw1; c(1201,u=1)=b1+b2+sw1+w1; c(2101,u=1)=(2b1)+sw1+w1; c(3010,u=2)=(2b1)+sw2+w2.

APPENDIX C: OPTIMALITY EQUATIONS OF C2 PROCESSING UNIT

$V0001=c0001+(\alpha \lambda V1001)+(\alpha v_1 V0011)+(\alpha \gamma_2 V0000)$ $+(\alpha(\mu_1 + \mu_2 + \gamma_1 + v_2)V0001);$

$V1010=min((c(1010,u=0)+(\alpha \lambda V2010)+(\alpha \gamma_1 V1000)+(\alpha v_2 V1011)+(\alpha(\mu_1 + \mu_2 + \gamma_2 + v_1)V1010)),(c(1010,u=2)+(\alpha \mu_1 V0010)+(\alpha(\lambda + \mu_2 + v_1 + v_2 + \gamma_1 + \gamma_2)V1010)));$

$V0010=c0010+(\alpha \lambda V1010)+(\alpha \gamma_1 V0000)+(\alpha v_2 V0011)+(\alpha(\mu_1 + \mu_2 + \gamma_2 + v_1)V0010);$

$V0011=c0011+(\alpha \lambda V1011)+(\alpha \gamma_1 V0001)+(\alpha \gamma_2 V0010)+(\alpha(\mu_1 + \mu_2 + v_1 + v_2)V0011);$

$V2010=min((c(2010,u=0)+(\alpha \lambda V3010)+(\alpha \gamma_1 V2000)+(\alpha v_2 V2011)+(\alpha(\mu_1 + \mu_2 + \gamma_2 + v_1)V2010)),(c(2010,u=2)+(\alpha \mu_1 V1010)+(\alpha(\lambda + \mu_2 + v_1 + v_2 + \gamma_1 + \gamma_2)V2010)));$

$V0101=min((c(0101,u=0)+(\alpha v_1 V0111)+(\alpha \gamma_2 V0100)+(\alpha \lambda V1101)+(\alpha(\mu_1 + \mu_2 + \gamma_1 + v_2)V0101)),(c(0101,u=1)+(\alpha \mu_2 V0001)+(\alpha(\lambda + \mu_1 + v_1 + v_2 + \gamma_1 + \gamma_2)V0101)));$

$V0201=min((c(0201,u=0)+(\alpha \lambda V1201)+(\alpha v_1 V0211)+(\alpha \gamma_2 V0200)+(\alpha(\mu_1 + \mu_2 + \gamma_1 + v_2)V0201)),(c(0201,u=1)+(\alpha \mu_2 V0101)+(\alpha(\lambda + \mu_1 + v_1 + v_2 + \gamma_1 + \gamma_2)V0201)));$

$V3010=min((c(3010,u=0)+(\alpha \gamma_1 V3000)+(\alpha v_2 V3011)+(\alpha(\lambda + \mu_1 + \mu_2 + v_1 + \gamma_2)V3010)),(c(3010,u=2)+(\alpha \mu_1 V2010)+(\alpha(\lambda + \mu_2 + v_1 + v_2 + \gamma_1 + \gamma_2)V3010)));$

$V0310=c0310+(\alpha \gamma_1 V0300)+(\alpha v_2 V0311)+(\alpha \mu_2 V0210)+(\alpha(\lambda + \mu_1 + v_1 + \gamma_2)V0310);$

$V1210=c1210+(\alpha \gamma_1 V1200)+(\alpha v_2 V1211)+(\alpha \mu_2 V1110)+(\alpha(\lambda + \mu_1 + v_1 + \gamma_2)V1210);$

$V0110=c0110+(\alpha \lambda V1110)+(\alpha \gamma_1 V0100)+(\alpha v_2 V0111)+(\alpha \mu_2 V0010)+(\alpha(\mu_1 + v_1 + \gamma_2)V0110);$

$V0200=c0200+(\alpha \lambda V1200)+(\alpha \nu_1 V0210)+(\alpha \nu_2 V0201)$
$+(\alpha \mu_2 V0100)+(\alpha (\mu_1+\gamma_1+\gamma_2)V0200);$

$V0111=c0111+(\alpha \lambda V1111)+(\alpha \gamma_1 V0101)+(\alpha \gamma_2 V0110)$
$+(\alpha (\mu_1+\mu_2+\nu_1+\nu_2)V0111);$

$V0210=c0210+(\alpha \lambda V1210)+(\alpha \gamma_1 V0200)+(\alpha \nu_2 V0211)$
$+(\alpha \mu_2 V0110)+(\alpha (\mu_1+\nu_1+\gamma_2)V0210);$

$V0300=c0300+(\alpha \nu_1 V0310)+(\alpha \nu_2 V0301)+(\alpha \mu_2 V0200)$
$+(\alpha (\lambda+\mu_1+\gamma_1+\gamma_2)V0300);$

$V1110=c1110+(\alpha \lambda V2110)+(\alpha \gamma_1 V1100)+(\alpha \nu_2 V1111)$
$+(\alpha \mu_2 V1010)+(\alpha (\mu_1+\nu_1+\gamma_2)V1110);$

$V0211=c0211+(\alpha \lambda V1211)+(\alpha \gamma_1 V0201)+(\alpha \gamma_2 V0210)$
$+(\alpha (\mu_1+\mu_2+\nu_1+\nu_2)V0211);$

$V2110=c2110+(\alpha \gamma_1 V2100)+(\alpha \nu_2 V2111)+(\alpha \mu_2 V2010)$
$+(\alpha (\lambda+\mu_1+\nu_1+\gamma_2)V2110);$

$V3001=c3001+(\alpha \nu_1 V3011)+(\alpha \gamma_2 V3000)+(\alpha \mu_1 V2101)$
$+(\alpha (\lambda+\mu_2+\nu_2+\gamma_1)V3001);$

$V0311=c0311+(\alpha \gamma_1 V0301)+(\alpha \gamma_2 V0310)+(\alpha (\lambda+\mu_1+$
$\mu_2+\nu_1+\nu_2)V0311);$

$V1211=c1211+(\alpha \gamma_1 V1201)+(\alpha \gamma_2 V1210)+(\alpha (\lambda+\mu_1+$
$\mu_2+\nu_1+\nu_2)V1211);$

$V2111=c2111+(\alpha \gamma_1 V2101)+(\alpha \gamma_2 V2110)+(\alpha (\lambda+\mu_1+$
$\mu_2+\nu_1+\nu_2)V2111);$

$V3011=c3011+(\alpha \gamma_1 V3001)+(\alpha \gamma_2 V3010)+(\alpha (\lambda+\mu_1+$
$\mu_2+\nu_1+\nu_2)V3011);$

$V0100=c0100+(\alpha \lambda V1100)+(\alpha \nu_1 V0110)+(\alpha \nu_2 V0101)$
$+(\alpha \mu_2 V0000)+(\alpha (\mu_1+\gamma_1+\gamma_2)V0100);$

$V2011=c2011+(\alpha \lambda V3011)+(\alpha \gamma_1 V2001)+(\alpha \gamma_2 V2010)$
$+(\alpha (\mu_1+\mu_2+\nu_1+\nu_2)V2011);$

$V2101=min((c(2101,u=0)+(\alpha \nu_1 V2111)+(\alpha \gamma_2 V2100)+($
$\alpha (\lambda+\mu_1+\mu_2+\gamma_1+\nu_2)V2101)),(c(2101,u=1)+(\alpha \mu_2 V20$
$01)+(\alpha (\lambda+\mu_1+\nu_1+\nu_2+\gamma_1+\gamma_2)V2101)));$

$V0301=min((c(0301,u=0)+(\alpha \nu_1 V0311)+(\alpha \gamma_2 V0300)+($
$\alpha (\lambda+\mu_1+\mu_2+\gamma_1+\nu_2)V0301)),(c(0301,u=1)+(\alpha \mu_2 V02$
$01)+(\alpha (\lambda+\mu_1+\nu_1+\nu_2+\gamma_1+\gamma_2)V0301)));$

$V1201=min((c(1201,u=0)+(\alpha \nu_1 V1211)+(\alpha \gamma_2 V1200)+($
$\alpha (\lambda+\mu_1+\mu_2+\gamma_1+\nu_2)V1201)),(c(1201,u=1)+(\alpha \mu_2 V11$
$01)+(\alpha (\lambda+\mu_1+\nu_1+\nu_2+\gamma_1+\gamma_2)V1201)));$

$V1200=c1200+(\alpha \mu_1 V0300)+(\alpha \nu_1 V1210)+(\alpha \nu_2 V1201)$
$+(\alpha \mu_2 V1100)+(\alpha (\lambda+\gamma_1+\gamma_2)V1200);$

$V2001=c2001+(\alpha \lambda V3001)+(\alpha \nu_1 V2011)+(\alpha \gamma_2 V2000)$
$+(\alpha \mu_1 V1101)+(\alpha (\mu_2+\gamma_1+\nu_2)V2001);$

$V2100=c2100+(\alpha \mu_2 V2000)+(\alpha \nu_1 V2110)+(\alpha \nu_2 V2101)$
$+(\alpha \mu_1 V1200)+(\alpha (\lambda+\gamma_1+\gamma_2)V2100);$

$V3000=c3000+(\alpha \nu_1 V3010)+(\alpha \nu_2 V3001)+(\alpha \mu_1 V2100)$
$+(\alpha (\lambda+\mu_2+\gamma_1+\gamma_2)V3000);$

$V1111=c1111+(\alpha \lambda V2111)+(\alpha \gamma_1 V1101)+(\alpha \gamma_2 V1110)$
$+(\alpha (\mu_1+\mu_2+\nu_1+\nu_2)V1111);$

$V1101=min((c(1101,u=0)+(\alpha \lambda V2101)+(\alpha \nu_1 V1111)+($
$\alpha \gamma_2 V1100)+(\alpha \mu_1 V0201)+(\alpha (\mu_2+\gamma_1+\nu_2)V1101)),(c(1$
$101,u=1)+(\alpha \mu_2 V1001)+(\alpha (\lambda+\mu_1+\nu_1+\nu_2+\gamma_1+\gamma_2)V11$
$01)));$

$V1011=c1011+(\alpha \lambda V2011)+(\alpha \gamma_1 V1001)+(\alpha \gamma_2 V1010)$
$+(\alpha (\mu_1+\mu_2+\nu_1+\nu_2)V1011);$

$V2000=c2000+(\alpha \lambda V3000)+(\alpha \nu_1 V2010)+(\alpha \nu_2 V2001)$
$+(\alpha \mu_1 V1100)+(\alpha (\mu_2+\gamma_1+\gamma_2)V2000);$

$V1100=c1100+(\alpha \lambda V2100)+(\alpha \nu_1 V1110)+(\alpha \nu_2 V1101)$
$+(\alpha \mu_1 V0200)+(\alpha \mu_2 V1000)+(\alpha (\gamma_1+\gamma_2)V1100);$

$V1001=c1001+(\alpha \lambda V2001)+(\alpha \nu_1 V1011)+(\alpha \gamma_2 V1000)$
$+(\alpha \mu_1 V0101)+(\alpha (\mu_2+\gamma_1+\nu_2)V1001);$

$V1000=c1000+(\alpha \lambda V2000)+(\alpha \nu_1 V1010)+(\alpha \nu_2 V1001)$
$+(\alpha \mu_1 V0100)+(\alpha (\mu_2+\gamma_1+\gamma_2)V1000);$

$V0000=c0000+(\alpha \lambda V1000)+(\alpha \nu_1 V0010)+(\alpha \nu_2 V0001)$
$+(\alpha (\mu_1+\mu_2+\gamma_1+\gamma_2)V0000);$

REFERENCES

Arena. Rockwell Software, Inc. Academic Version 7.01.00. 2004.

Cassandras, C. G., and S. Lafortune. 1999. *Introduction to Discrete Event Systems*. Chapter 8. Kluwer Academic Publishers.

Cruz, J. B. Jr., M. A. Simaan, A. Gacic, H. Jiang, B. Letellier, M. Li, and Yong Liu. 2001. Game-theoretic mod-

eling and control of a military air operation. In *IEEE Transactions on Aerospace and Electronic Systems.* 37: 1393-1405.

Deb, R. K., and R. F. Serfozo. 1973. Optimal control of batch service queues. In *Adv. Appl. Prob*. 5: 340–361.

Kelton, W. D., R. P. Sadowski and D. T. Sturrock. 2004. *Simulation with Arena. 3rd Edition.* McGraw-Hill.

Law, A. M. and W. D. Kelton. 2000. *Simulation Modeling and Analysis. 3rd edition.* McGraw Hill.

Pepyne, D. L., and C. G. Cassandras. 1997. Optimal dispatching control for elevator systems during uppeak traffic. In *IEEE Transactions on Control Systems Technology*, 5(6): 629-643.

Wohletz, J. M., D. A. Castanon, and M. L. Curry. Closed-loop control for joint air operations. In *Proc. American Control Conference* (ACC 2001).

Wu, N. E., and T. Busch. 2004. Operational reconfigurability in command and control. In *Proc. American Control Conference* (ACC 2004).

Wu, N. E., and T. Busch. 2004. An example of Supervisory Control in C2. In *Proc. IEEE Conference on Decision and Control* (CDC 2004*)*.

Wu, N. E., S. Thavamani. 2006. Effect of Acknowledgement on Performance of a Fault-Tolerant Wireless Network. To appear in *5th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes* (SAFEPROCESS 2006).

AUTHOR BIOGRAPHY

SUDHA THAVAMANI is a doctoral student in the department of Electrical and Computer engineering at Binghamton University. She previously completed her master's degree in Electrical Engineering from Binghamton University. She holds the bachelors degrees in Instrumentation Engineering from Anna University, Chennai, India and Physics from University of Madras, Chennai, India. Her research interests include fault tolerant control, modeling and simulation. She is a member of Instrumentation Systems and Automation Society (ISA), student member of the IEEE, IEEE Control Systems Society and Women in Control Committee of IEEE Control Systems Society. Her e-mail address is <sudha@binghamton.edu>.