Analysis of Manufacturing Blocking Systems with Network Calculus ⋆

Amit Bose [a,1] Xiaoyue Jiang [b] Bin Liu [c] Gang Li [a]

[a] *School of Mathematics and Statistics, Carleton University*

[b] *Department of Industrial Engineering, Louisiana State Univesity*

[c] *Institute of Applied Mathematics, Academy of Mathematics and System Sciences, Chinese Academy of Sciences*

Abstract

In this paper, manufacturing blocking (MB) system is studied from the Network Calculus perspective. By dominating MB by a window flow controller (WFC), we obtain the service curve of the system for both instantaneous and non-instantaneous cases. The explicit expression of the system service curve further leads to the optimal allocation of the buffer sizes in order to guarantee ideal system service curve (i.e. the service curve for the system with unlimited buffers between servers). This allocation is more efficient than that based on the guarantees of individual service curve. In addition, the simulation of NetCal systems is developed based on the duality between arrival curve and strict service curve. The concept of Workload Regulation is introduced to enforce a service curve of the rate-latency form. This construction provides the service regulator in the same way as the leaky bucket enforces the arrival curve. Simulation experiments are conducted to show the tightness of the theoretical bounds obtained using Network Calculus.

Key words: Manufacturing Blocking; Window Flow Control; Network Calculus; Service curve; Simulation; Performance Bound; Buffer allocation; QoS

1 Introduction

For a tandem system with limited buffer capacity, overflow may occur if a packet, after completion of its service, could not find space at the downstream

⋆ This work is partially supported by NSERC, Alcatel, NCIT and MITACS.

[1] Corresponding author: abose@math.carleton.ca

buffer. Manufacturing blocking (MB) mechanisms may be used to avoid overflow, and to minimize the amount of wasted effort should drop of a packet be required.

The traditional analyses of MB is based on Markov Chain approach ([1], [9] and [14]). However, existing results on tandem queues are mainly confined to analysis of backlog, even though the expected delay time can be obtained by the aid of Little's formula. The calculation of the probability distribution of the delay time is complicated even for a 2-node system, which may involve cumbersome derivations of functional Laplace-transforms. Consequently, optimization of buffer allocation by traditional methods is hard to carry out. Finally, these methods focus on the steady-state behaviors of the systems, which do not allow for worst case analysis.

The theory of Network Calculus (NetCal) offers a new alternative to stochastic queueing theory, with special emphasis on worst case analysis, or in other words, performance bound analysis. Based on (min, +) algebra, NetCal has been developed to handle deterministic queueing systems in communication networks. Pioneering works in this area were undertaken by [3,4], [6,7], [12] among others, and the key contributions, up to now, have been well presented in [5] and in [13].

A variety of performance analysis issues have been successfully treated by NetCal, which includes: offering a common language for packet schedulers; computing delay bounds used in the IETF guaranteed service protocol; defining deterministic effective bandwidth, and video traffic smoothing, just to name a few.

The main theoretical advantages of NetCal is that it supplies hard performance bounds needed for Quality of Service(QoS) guarantees.

The concatenation theorem in NetCal shows that the global service curve of a system consisting of servers in tandem is the (min, +) convolution of individual service curves. This result enables one to connect multiple nodes into a network which still possesses a NetCal description. However, an implicit assumption of the concatenation result is that the buffers between consecutive nodes are large enough to avoid blockage of service. Due to finiteness of the buffer sizes between tandem servers, the service curve obtained from convolution may not be guaranteed.

In communications network engineering, the window flow control (WFC) mechanism is often used to coordinate the packet processing between servers in tandem. This control mechanism has a nice NetCal representation. By observing the connection between WFC and MB, we are able to characterize MB from the NetCal perspective.

The rest of the paper is organized as follows.

In Section 2, we prove that the throughput a WFC system is dominated by

corresponding MB system where the buffer sizes in the MB equal the window sizes in the WFC. In Section 3, we obtain the global service curve for Non-instantaneous N-server WFC system. This allows us to characterize MB system's worst case performance based on NetCal methodology. In addition, with the explicit formula of the global service curve, we find the optimal buffer allocation schemes which guarantee the global service curve, individual service curves, and non-occurrence of blocking. It turns out these buffer allocations are different. Finally, we introduce a notion of Workload Regulation for the construction of NetCal simulation. Next, simulation experiments are conducted to illustrate the tightness of theoretical bounds.

2 Dominance of MB by WFC

In this section, we study two flow control mechanisms between queueing servers: manufacturing blocking (MB), and window flow control (WFC). We assume that all servers are FIFO. We prove that the delays in MB system are dominated from above by the WFC system. Therefore, the service curve guaranteed by WFC, to be derived in the next section, is also guaranteed by the MB system.

Definition 2.1 ([2]) Manufacturing blocking (production blocking) *Consider a serial system of servers, where each server always serves a packet as long as there is a packet available for processing, and it is not "blocked". It is termed blocked if a packet with completed service cannot proceed to the downstream buffer because that buffer is full. The server is immediately unblocked as soon as the downstream buffer is available to receive a new packet.*

Let $A^X(n)$ be the arrival times in flow X, and $D_i^X(n)$ be the departure time of the nth packet from the server $i \leq K$. Suppose the system is empty at time zero. From definition, it is well-known ([2], p. 185, (5.39)) that the following recursive formula holds for manufacturing blocking system.

$$D_i^X(n) = [(D_{i-1}^X(n) \vee D_i^X(n-1)) + S_i(n)] \vee D_{i+1}^X(n-b_i) \tag{2.1}$$

for $n \geq 1;\ i \leq K$, where $D_i^X(n) = 0,\ n \leq 0;\ D_0^X(n) = A^X(n)$, and $D_{K+1}^X(n) = 0$, for all $n \geq 1$, and $b_i \geq 1$ is the output buffer of the ith server (one position at the server (i+1) is always included in this count) . The following monotone property is a direct consequence of the dynamic equation (2.1).

Lemma 2.2 1 (Monotone Property of MB) *Suppose two traffic flows, X and Y are passing through two identical MB systems and moreover the system is empty at time zero. Assume $A^X(n) \leq A^Y(n)$ are the arrival times of the nth packet in flows X and Y respectively. Then the departure times satisfy*

$$D_i^X(n) \leq D_i^Y(n),\ i \leq K, n \geq 1. \tag{2.2}$$

PROOF. Notice that the MB dynamic equation (2.1) contains only the order preserving operators, $\vee$ and $+$. It is clear that for the traffic flows with arrival times $A^X(n) \leq A^Y(n)$, the same order still holds for the the departure process. That is, $D^X(n) \leq D^Y(n)$. This completes the proof of Lemma 2.2. $\square$

Remark 2.3 *In Lemma 2.2, the "identical" is defined as that it is almost sure that the service times of the nth packet in both flows at any server i in the system are equal, i.e., $S_i^X(n) = S_i^Y(n) = S_i(n)$. The comparison is at the sample path level, and the systems comparison should be understood in the stochastic sense. The dominance property to be shown in Theorem 2.5 should also be understood in the same way.*

We formulate the manufacturing blocking mechanism based on WFC. The window flow controller limits the amount of data admitted into the network so that the total backlog in the network is less than or equal to a constant W (Figure 1, see also [5], P.82).

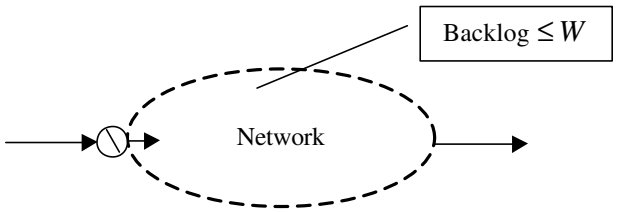


Fig. 1. Window flow Control Mechanism.

Definition 2.4 ([13]) (Window Flow Control) *A packet, after its arrival, is allowed to enter the network at time t if $q(t)$, the total number of packets in the network, is less than the window size W immediately after its entry to the network.*

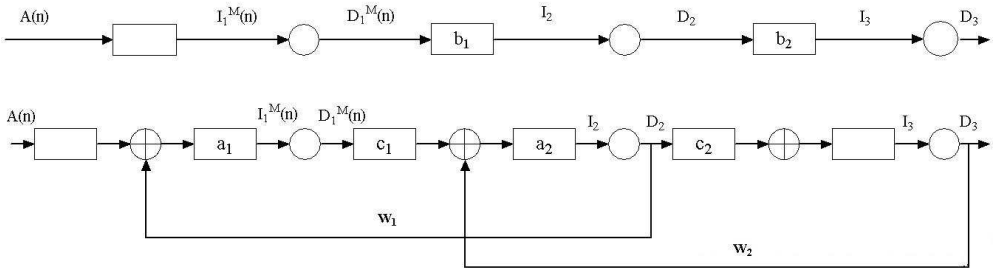


Fig. 2. MB and corresponding WFC

The second diagram in Figure 2 is one physical implementation of the WFC. This differs from the conceptual diagram on Figure 1 in that a buffer is added to each of the controller's entry points. These buffers are the output buffers for the up-stream servers. They are needed because based on the WFC assumption, the up-stream sever maintains the same service rate as the one

with no buffer constraint. Therefore, the output buffer is needed to prevent the blocking within each window.

With this realization of the WFC, we need to allocate the buffers as follows

$$a_1 = W_1,\ c_1 = W_1;\ a_i = W_i \wedge W_{i+1},\ c_i = W_i\,(i \geq 2) \tag{2.3}$$

For two identical sequences of servers, one controlled by MB and the other by WFC, if the buffer size b_i equals the window size W_i, then the Theorem 2.5 states that the throughput from each server in the MB system is larger than that of its counterpart in the corresponding WFC system.

Theorem 2.5 (Dominance of WFC) *Let K be the number of the servers in the system. For $i \leq K$, denote $I_i^W(n), I_i^M(n), D_i^W(n), D_i^M(n)$ as the arrival and departure time of the nth packet at the ith server in the WFC, and MB systems, respectively. Then for any $n \geq 1$,*

$$D_i^W(n) \geq D_i^M(n). \tag{2.4}$$

PROOF. For $K = 1$, there is neither WFC nor MB control mechanism. Therefore $D_1^M = D_1^W$, and the equation (2.4) is true.

Assume (2.4) is true for $K > 1$, and for any $n \geq 1$.

We need to prove that for any MB system with $K + 1$ servers, the result is true.

We proceed with mathematical induction on n, the sequence number of a packet in the traffic flow.

For $n = 1$, and server i, $1 \leq i \leq K + 1$, we have

$$D_i^W(1) = D_i^M(1), \tag{2.5}$$

as the first packet is never blocked in either of the MB and WFC systems.

Let $n = m$, and assume for any packet j, $1 \leq j \leq m$, and server i, $1 \leq i \leq K + 1$, that

$$D_i^W(j) \geq D_i^M(j). \tag{2.6}$$

This is actually the assumption stated in Lemma 2.6. Hence,

$$D_1^M(m + 1) \leq D_1^W(m + 1). \tag{2.7}$$

Notice that the downstream K server sub-system forms a MB(WFC) system by itself. The departing packet from the server one at time $D_1^M(n)(D_1^W(n))$ becomes the input packet to this K server MB (WFC) sub-system. From the induction assumption applied to the sub-system, and the monotone property

of MB in Lemma 2.2, one has

$$D_i^W(m+1) \geq D_i^M(m+1) \text{ for } 2 \leq i \leq (K+1). \tag{2.8}$$

Now (2.7) and (2.8) complete the proof of the theorem. □

Lemma 2.6 *If for packet j, $1 \leq j \leq m$, and server i, $1 \leq i \leq K+1$, we have*

$$D_i^W(j) \geq D_i^M(j), \tag{2.9}$$

then

$$D_1^W(m+1) \geq D_1^M(m+1). \tag{2.10}$$

PROOF. From the dynamics of MB, (2.1), we have

$$D_1^M(m+1) = [A(m+1) \vee D_1^M(m) + S_1(m+1)] \vee D_2^M(m+1-b_1). \tag{2.11}$$

Note that $D_1^W(m+1) \geq A(m+1) + S_1(m+1)$. We need only to prove

$$\begin{aligned} D_1^W(m+1) &\geq D_1^M(m) + S_1(m+1) \\ D_1^W(m+1) &\geq D_2^M(m+1-b_1). \end{aligned} \tag{2.12}$$

The first inequality in (2.12) follows from assumption (2.6) for $j = m$, i.e., $D_1^W(m) \geq D_1^M(m)$, and $D_1^W(m+1) \geq D_1^W(m) + S_1(m+1)$.

For the second one, note that $b_1 \geq 1$, then we have $(m+1-b_1) \leq m$, and $D_2^W(m+1-b_1) \geq D_2^M(m+1-b_2)$ from (2.6). Therefore, we need only to prove

$$D_1^W(m+1) \geq D_2^W(m+1-b_1) \tag{2.13}$$

This is automatically true from the definition of WFC. Because if it were not true, then, $D_1^W(m+1) < D_2^W(m+1-b_1)$. For any time T, such that $D_1^W(m+1) < T < D_2^W(m+1-b_1)$, then the packets with sequence numbers $(m+1-b_1), ..., (m+1)$ are in the first window. And the total number of the packets equals to $b_1 + 1$. Recall we have window flow controller for the first window with size $W_1 = b_1$, so it is a contradiction, which completes the proof of Lemma 2.6. □

3 NetCal Characterization of WFC

Let us now recall basic notations and results from the NetCal theory. The reader may consult [13] for further details.

Definition 3.1 (Convolution) *Let f, g be increasing non-negative function on $[0, +\infty)$. The $(min, +)$-convolution operator $\otimes$ is defined as*

$$f \otimes g(t) = \inf_{0 \le s \le t} \{f(s) + g(t-s)\}. \tag{3.1}$$

The n-fold convolution of f with itself is denoted by

$$f^{(n)} = f \otimes f \otimes ... \otimes f. \tag{3.2}$$

Definition 3.2 (De-convolution) *The de-convolution operator $\oslash$ is defined as follows*

$$f \oslash g(t) = sup_{s \ge 0}\{f(s+t) - g(s)\}. \tag{3.3}$$

Definition 3.3 (Sub-Additive Closure) *The sub-additive closure is defined as*

$$\overline{f} = \delta_0 \wedge (\bigwedge_{n=1}^{\infty} f^{(n)}) \tag{3.4}$$

Definition 3.4 (Arrival Curve) *We say that a flow x is constrained by arrival curve α if and only if for all $s \le t$:*

$$x(t) - x(s) \le \alpha(t-s). \tag{3.5}$$

Equivalently, for all $t \ge 0$, we have $x(t) \le (\alpha \otimes x)(t)$. We also say that x has α as an arrival curve, or x is α-smooth.

Definition 3.5 (Service Curve) *For a flow through service node S with input and output function $x(t)$, and $y(t)$, we say that S offers to the flow a service curve β if and only if for all $t \ge 0$, there exists some $t_0 \le t$, such that*

$$y(t) - x(t_0) \ge \beta(t - t_0). \tag{3.6}$$

Again, using $\otimes$ operation, it is equivalent to $y(t) \ge (\beta \otimes x)(t)$.

Definition 3.6 (Strict Service Curve) *A system S offers a strict service curve β if during any backlogged period of duration u, the output of the flow is at least $\beta(u)$.*

It is known that if a node offers β as a strict service curve to a flow, then it offers β as a service curve to this flow.

Definition 3.7 (Maximum Service Curve) *Consider a system S and a flow through S with input and output function x and y. We say that S offers to the flow a maximum service curve γ if and only if γ is wide-sense increasing function and $y \le x \otimes \gamma$.*

Lemma 3.8 (Node Concatenation) ([13] Theorem 1.4.6, p.34) *The concatenation of a series of nodes with service curves $\beta_i, i = 1, .., K$, guarantees a system service curve β_g*

$$\beta_g = \beta_1 \otimes \beta_2 \otimes ... \otimes \beta_K. \tag{3.7}$$

Remark 3.9 *The concatenation result stated in Lemma 3.8 implicitly assumes that the buffers between servers are infinite. For systems with finite buffer sizes and MB control mechanism, we will find the global service curve of the system, which will help us determine in the next section the minimal size of each buffer, such that a) no blocking occurs; that is, b) individual service curves β_i are maintained; or equivalently, c) the ideal system service curve $\beta_1 \otimes \cdots \otimes \beta_N$ is maintained. That is, the system service curve does not improve by making the buffers larger than certain threshold values.*

The dominance result shown in Section 2 implies that MB system guarantees the same global service curve, consequently, which enables us to derive MB bounds based on the NetCal bounds of WFC.

Moreover, in real life, only non-instantaneous communications between servers are possible. So we will first formulate WFC for two-server systems in the non-instantaneous case. The result is then generalized for the N-sever case, which includes instantaneous case.

3.1 2-Server WFC

We now consider the case when both the transmission and the acknowledgement between the server 1 and the server 2 take non-negligible time. The results in this section have been established in [8] under the assumption of "strict" service curves, as defined in that paper. Their notion is stronger than the service curve concept and is weaker than the strict service curve concept that is currently in use (see Definition 3.6). Here we provide an alternative proof which significantly simplifies the original proof, and also generalizes the original result to the case of general service curves.

Suppose that traffic departs the first server and feeds a "network element", N^f, which serves traffic in a FIFO manner and feeds the second buffer.

As traffic from the original source departs the second buffer, acknowledgements are correspondingly generated by the second server and sent back to the first server via a network element N^b. The network element N^b operates in a FIFO manner.

The following Figure 3 illustrates the configuration of the system, and Theorem 3.10 gives the actual guaranteed service curve of server 1 under window flow control.

Theorem 3.10 *Consider a system with two servers under WFC with window size W. Denote the guaranteed service curve of server i as $\beta_i (i = 1, 2)$. The network elements N^f and N^b guarantee the service curve S^f and S^b, respectively. Define $S_{loop} = \beta_1 \otimes S^f \otimes \beta_2 \otimes S^b$ and β_g as the global service curve of*

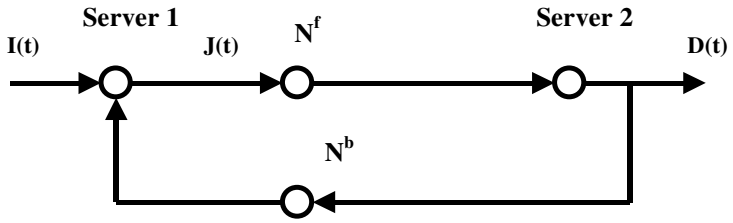


Fig. 3. Window Flow Control with Non-instantaneous Feedback.

the system. Then the server one guarantees the service curve β_1'*, where*

$$\begin{aligned}\beta_1' &= \min_{m\in Z^+}\{\beta_1 \otimes S_{loop}^{(m)} + mW\} \\ &= \beta_1 \otimes \overline{S_{loop} + W} \\ &= \beta_1 \otimes \overline{\beta_1 \otimes \beta_2 \otimes S^f \otimes S^b + W},\end{aligned} \tag{3.8}$$

and

$$\beta_g = \beta_1 \otimes \beta_2 \otimes S^f \otimes \overline{S_{loop} + W}. \tag{3.9}$$

Here Z^+ *is the set of non-negative integers, and* $\overline{\beta}$ *is the sub-additive closure of* β*.*

PROOF. Let A be the arrival process, I be the effective input, i.e., $I(t)$ is the number of arrivals that actually enter the window by time t. Let D be the output process, and D' the feedback process.

Then the effective input is determined by the following equation:

$$I(t) = \min\{A(t), D'(t) + W\} \tag{3.10}$$

The network offers service curve $\beta = \beta_1 \otimes \beta^f \otimes \beta_2$ for the packet flow. Clearly, $\beta_{loop} = \beta \otimes S^b$. Then

$$\begin{aligned}D &\geq I \otimes \beta \\ &= [A \wedge (D' + W)] \otimes \beta \\ &= (A \otimes \beta) \wedge (D' \otimes (\beta + W)) \\ &= (A \otimes \beta) \wedge (D \otimes (\beta \otimes \beta^b + W)) \\ &= (A \otimes \beta) \wedge (D \otimes (\beta_{loop} + W)).\end{aligned} \tag{3.11}$$

By Theorem 2.1.6 ([5]),

$$D \geq (A \otimes \beta) \otimes \overline{\beta_{loop} + W}. \tag{3.12}$$

Consequently,

$$\begin{aligned} I &= A \wedge (D' + W) \\ &\geq A \wedge ((A \otimes \beta) \otimes \overline{\beta_{loop} + W} \otimes S^b + W) \\ &= A \wedge (A \otimes (\beta_{loop} + W) \otimes \overline{\beta_{loop} + W}) \\ &= A \otimes \overline{\beta_{loop} + W}. \end{aligned} \tag{3.13}$$

Finally, for the output from server 1, denoted by $J(t)$, we have

$$\begin{aligned} J &\geq I \otimes \beta_1 \\ &= A \otimes \beta_1 \otimes \overline{\beta_{loop} + W}. \end{aligned} \tag{3.14}$$

which completes the proof of (3.8).

Equation (3.9) follows by the concatenation result for service nodes, Lemma 3.1 ([13], Theorem 1.4.6, p.34). □

As a corollary, we have:

Corollary 3.11 ([13]) *For the instantaneous case, i.e., $S^f = S^b = \delta_0$, then server 1 in the 2-server WFC system has service curve β_1', such that*

$$\beta_1' = \beta_1 \otimes \overline{\beta_1 \otimes \beta_2 + W}. \tag{3.15}$$

The global service curve is $\beta_1' \otimes \beta_2 = \beta_1 \otimes \beta_2 \otimes \overline{\beta_1 \otimes \beta_2 + W}$.

Corollary 3.12 *For the instantaneous case, the 2-server MB system with buffer size b guarantees the same service curve $\beta_1' \otimes \beta_2$ as in the corresponding WFC system with window size W if $b = W$.*

Remark 3.13 *Interestingly, the leaky bucket regulator can be viewed as a special case of manufacturing blocking, where the buffer size is the bucket size, i.e., $W = b$, $\beta_1 = \delta_0$ and $\beta_2 = rt$, i.e., it is a constant bit rate server with the rate same as the leaky rate, i.e., $\beta(t) = rt$. By* (3.13)*,*

$$\begin{aligned} I &\geq A \otimes \overline{S_{loop} + W} \\ &= A \otimes \overline{rt + b} \\ &= A \otimes [(rt + b) \wedge \delta_0] \\ &= A \otimes \gamma_{r,b}. \end{aligned} \tag{3.16}$$

Here, A is interpreted as the arbitrary arrival process, I is the regulated traffic, and $\gamma_{r,b}$ becomes the arrival curve enforced by the leaky bucket.

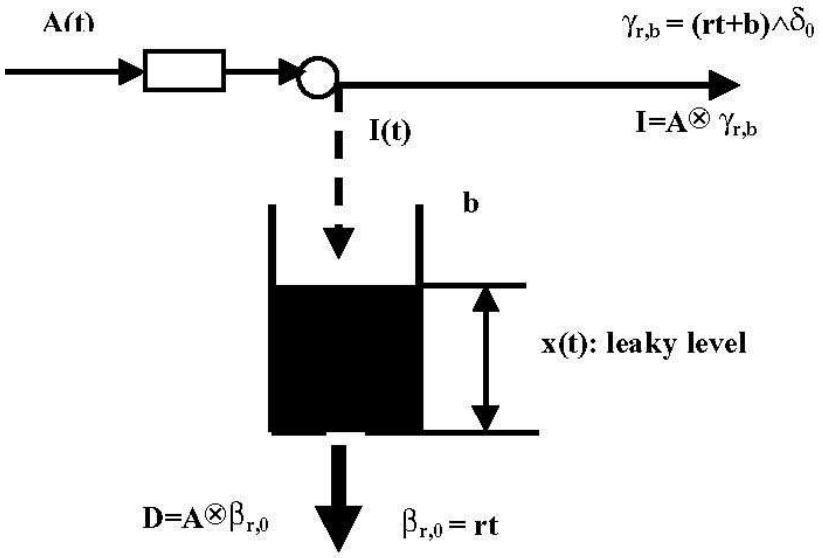


Fig. 4. Leaky Bucket from a NetCal Perspective

On the other hand, denote $D(t)$ as the output from the bucket. Then $I(t) \leq D(t) + b$. It is easy to see that

$$D \leq D \otimes \beta_{r,0} \leq I \otimes \beta_{r,0} \leq A \otimes \beta_{r,0}. \tag{3.17}$$

Therefore, $I \leq A \otimes \beta_{r,0} + b$, and $I \leq A \otimes \gamma_{r,b}$, as $I(0) = 0$. Combined with (3.16), we have $I = A \otimes \gamma_{r,b}$.

Similarly, for $D(t)$, we have

$$\begin{aligned} D &\geq A \otimes \beta_1 \otimes \beta_2 \otimes \overline{S_{loop} + W} \\ &= A \otimes \beta \otimes \overline{rt + b} \\ &= A \otimes \beta_{r,0}. \end{aligned} \tag{3.18}$$

Combined with (3.17), we have $D = A \otimes \beta_{r,0}$.

3.2 N-Server WFC

Consider the system consisting of $N > 2$ servers in series, all servers except the last one are controlled hop-by-hop by window flow control mechanism (Figure 2). The server i allows a data packet in buffer i to enter for service only when the backlogged data packets between server i and server $i + 1$ (including ones in service) is less than W_{i+1}, otherwise the server i does not commence a new service.

Now we consider the system that consists of $N > 2$ servers in series, and each server except the last one is control hop-by-hop window flow control

mechanism. The key result in Theorem 3.14 is first shown in ([5], P. 83) for the instantaneous case. Here we provide an alternative proof with some extra results for the individual service curves.

Denote the individual service curve as β_i, and assume the network elements N_i^f and N_i^b guarantees the service curve S_i^f and S_i^b, respectively. Denote $S_{loop}^i = \beta_{i+1} \otimes \beta_i \otimes S_i^f \otimes S_i^b$, and $S_{loop}^{'i} = \beta'_{i+1} \otimes \beta_i \otimes S_i^f \otimes S_i^b$,

From (3.8), we have for $1 \leq i \leq N-1$,

$$\begin{aligned} \beta'_i &= \beta_i \otimes \overline{S_{loop}^{'i} + W_i} \\ \beta'_{N-1} &= \beta_{N-1} \otimes \overline{S_{loop}^{N-1} + W_{N-1}} \end{aligned} \tag{3.19}$$

Denote the global service curve as β_g. From (3.19) and the concatenation result, we have that

$$\begin{aligned} \beta_g &= \beta'_1 \otimes S_1^f \otimes \beta'_2 \otimes S_2^f \otimes ... \otimes \beta'_{N-1} \otimes S_{N-1}^f \otimes \beta_N \\ &= \bigotimes_{i=1}^{N-1} (\beta_i \otimes S_i^f \otimes \overline{(S_{loop}^{'i} + W_i)}) \otimes \beta_N. \end{aligned} \tag{3.20}$$

Theorem 3.14 *For* $1 \leq i \leq N-1$,

$$\beta'_i \geq \beta_i \otimes \bigotimes_{j=i}^{N-1} \overline{(S_{loop}^j + W_j)} \tag{3.21}$$

and β_g *satisfies*

$$\beta_g = \bigotimes_{i=1}^{N-1} (\beta_i \otimes S_i^f \otimes \overline{(S_{loop}^i + W_i)}) \otimes \beta_N. \tag{3.22}$$

which is the guaranteed global service curve for the N-station manufacturing blocking system.

PROOF. First, we prove

$$\overline{S_{loop}^{'i} + W_i} \geq \bigotimes_{j=1}^{N-1} \overline{S_{loop}^j + W_i} \tag{3.23}$$

by backward induction from $(i+1)$ to i, beginning at $i = N-1$. Then (3.21) follows from (3.23) directly.

In the sequel, we prove the main part of the theorem. When $i = N-1$, we have $\beta'_N = \beta_N$. From Theorem 3.10, (3.23) holds.

From the induction assumption, (3.21) is true for any indices larger than i. Denote

$$B_i = S_{loop}^{'i} + W_i. \tag{3.24}$$

Therefore, (3.19) is rewritten as

$$\beta_i' = \beta_i \otimes \overline{B_i}, \tag{3.25}$$

and we have

$$\begin{aligned} \overline{B_i} &= \overline{(\beta_i \otimes S_i^f \otimes S_i^b + W_i) \otimes \beta_{i+1}'} \\ &= \overline{(\beta_i \otimes S_i^f \otimes S_i^b + W_i) \otimes \beta_{i+1} \otimes \overline{B_{i+1}}} \\ &\geq \overline{(S_{loop}^i + W_i)} \otimes \overline{B_{i+1}} \\ &\geq \bigotimes_{j=i}^{N-1} \overline{(S_{loop}^j + W_j)}. \end{aligned} \tag{3.26}$$

The first inequality is from $\overline{f \otimes \overline{h}} \geq \overline{f} \otimes \overline{h}$, and the last inequality uses the induction assumption.

From (3.23) and (3.25), we have

$$\begin{aligned} \beta_i' &\geq \beta_i \otimes \bigotimes_{j=i}^{N-1} \overline{(S_{loop}^j + W_j)} \\ &= \beta_i \otimes \overline{\bigwedge_{j=i}^{N-1} (S_{loop}^j + W_j)}. \end{aligned} \tag{3.27}$$

Finally, by (3.20),

$$\begin{aligned} \beta_g &= \beta_1' \otimes S_1^f \otimes \beta_2' \otimes S_2^f \otimes ... \otimes \beta_{N-1}' \otimes S_{N-1}^f \otimes \beta_N \\ &\geq \beta_1 \otimes S_1^f \otimes \overline{\bigwedge_{i=1}^{N-1} (S_{loop}^i + W_i)} \otimes \\ &\quad \beta_2 \otimes S_2^f \otimes \overline{\bigwedge_{i=2}^{N-1} (S_{loop}^i + W_i)} \otimes \\ &\quad ... \\ &\quad \beta_{N-1} \otimes S_{N-1}^f \otimes \overline{\bigwedge_{i=N-1}^{N-1} (S_{loop}^i + W_i)} \otimes \beta_N \\ &= \bigotimes_{i=1}^{N-1} (\beta_i \otimes S_i^f) \otimes \overline{\bigwedge_{i=1}^{N-1} (S_{loop}^i + W_i)} \otimes \beta_N \\ &= \bigotimes_{i=1}^{N-1} (\beta_i \otimes S_i^f \otimes \overline{(S_{loop}^i + W_i)}) \otimes \beta_N. \end{aligned} \tag{3.28}$$

From $\beta_i' \leq \beta_i$, and (3.20), we have

$$\beta_g \leq \bigotimes_{i=1}^{N-1} (\beta_i \otimes S_i^f \otimes \overline{(S_{loop}^i + W_i)}) \otimes \beta_N. \tag{3.29}$$

The combination of (3.29) and (3.28) completes the proof. □

As a special case, we have:

Lemma 3.15 *If the transmission and signalling between servers are instantaneous, then the system guarantees a global service curve*

$$\beta_g = (\bigotimes_{i=1}^{N} \beta_i) \otimes \bigotimes_{i=1}^{N-1} \overline{(\beta_i \otimes \beta_{i+1} + W_i)}. \tag{3.30}$$

In particular, assume

$$\beta_i = \beta_{r_i,t_i} = r_i[t - t_i]^+. \tag{3.31}$$

Denote $R = \wedge_{i=1}^{N} r_i$, and $T = \sum_{i=1}^{N} t_i$, then

$$\beta_g = \beta_{R,T} \otimes \overline{\wedge_{i=1}^{N-1}(\delta_{(t_i+t_{i+1})} + W_i)}. \tag{3.32}$$

4 Optimal Buffer Size Allocation

Next, we are going to determine bounds for $W_i, \forall\ 0 \leq i \leq (N-1)$ such that $\beta_g = \beta_{ideal}$, where

$$\beta_{ideal} = \otimes_{i=1}^{N-1}(\beta_i \otimes S_i^f) \otimes \beta_N. \tag{4.1}$$

Note that β_{ideal} is the system service curve when $W_{i+1} = \infty$. We are looking for a finite buffer allocation which guarantees β_{ideal} which equals the global service curve, β_g.

Theorem 4.1 (a). *The system guarantees the service curve β_{ideal} if $\forall 1 \leq i \leq N-1$,*

$$W_i \geq \sup_{t>0}[(\beta_{ideal} \oslash \beta_{ideal}) - S_{loop}^i](t). \tag{4.2}$$

(b). *The individual nodes guarantee the service curve β_i if $\forall 1 \leq i \leq N-1$,*

$$W_i \geq \sup_{t>0}[(\beta_i \otimes S_i^f) \oslash (\beta_i \otimes S_i^f) - S_{loop}^i](t), \tag{4.3}$$

(c). *Assume the traffic input is constrained by the arrival curve α. The minimal buffer size to guarantee that no blocking occurs at any server is*

$$W_i \geq \sup_{t>0}[(\alpha - \otimes_{k=1}^{i+1}\beta_k)](t), \tag{4.4}$$

PROOF. (a) To ensure $\beta_g = \beta_{ideal}$, we need for all $1 \leq i \leq N-1$,

$$\beta_{ideal} \otimes \overline{S_{loop}^i + W_i} \geq \beta_{ideal}. \tag{4.5}$$

As it is easy to see that $f \oslash g \leq h$ iff $f \leq g \otimes h$, it is equivalent to

$$\overline{S_{loop}^i + W_i} \geq \beta_{ideal} \oslash \beta_{ideal}. \tag{4.6}$$

Note that $\beta_{ideal} \oslash \beta_{ideal}$ is sub-additive, then (4.6) is equivalent to

$$\overline{S_{loop}^i + W_i} \geq \overline{\beta_{ideal} \oslash \beta_{ideal}}, \tag{4.7}$$

and we need only

$$S^i_{loop} + W_i \geq \beta_{ideal} \oslash \beta_{ideal}. \tag{4.8}$$

Now (4.2) follows immediately, and (a) is proved.

(b) The proof is almost identical to (a).

(c) The proof is a direct application of the results on output bound, and backlog bound (see [13], p. 28). In fact, the output from server i is bounded by $\alpha \oslash \otimes_{k=1}^{i} \beta_k$, and therefore the backlog is bounded by $\sup_{t\geq 0}[\alpha \oslash \otimes_{k=1}^{i} \beta_k - \beta_{i+1}](t)$, which equals to right hand side in (4.4). □

Remark 4.2 *(Special case)* $\beta_i = \beta_{r_i,t_i}$ *and* $S^f_i, S^b_i = \delta_0$. *We have* $\beta_{ideal} = \beta_{R,T}$, $\beta_{ideal} \oslash \beta_{ideal} = \beta_{R,0}$, *and* $\beta_i \otimes \beta_{i+1} = \beta_{r_i \wedge r_{i+1},(t_i+t_{i+1})}$. *Therefore,* $\forall n \geq 1$,

$$\sup_{t\geq 0}(\beta_{ideal} \oslash \beta_{ideal} - (\beta_i \otimes \beta_{i+1}))$$

$$= R(t_i + t_{i+1}) = (\textstyle\bigwedge_{i=1}^{N} r_i)(t_i + t_{i+1}).$$

Therefore, the ideal system service curve $\beta_{ideal} = \beta_{R,T}$ is guaranteed if $\forall\ 1 \leq i \leq N-1$,

$$W_{i+1} \geq (\bigwedge_{i=1}^{N} r_i)(t_i + t_{i+1}). \tag{4.9}$$

On the other hand, to guarantee individual nodes' service curves, it is easy to see that the following two condition must be satisfied:

$$r_{i+1} \geq r_i, \tag{4.10}$$

and

$$W_{i+1} \geq r_i(t_i + t_{i+1}). \tag{4.11}$$

Clearly, (4.10) together with (4.11) are much stronger than (4.9) alone, which implies that it is unnecessary to allocate sufficient buffer/bandwidth to ensure $\beta'_i = \beta_i$ to maintain the ideal system service curve. In addition, the buffer bounds in (a), (b) is not affected by the input traffic. However, the bounds to guarantee no-blocking in (c) depends on the input regulation α. This indicates that there is no need to guarantee non-blocking in order to guarantee the ideal service curves.

Finally, for the non-instantaneous case, with additional assumptions, it is possible to allocate the buffer sizes less than the corresponding window sizes without causing packet loss. Following the ideas similar to ([8] and [11]), we have the following result, which generalizes the results based on NetCal concepts.

Theorem 4.3 *Suppose that* N^b *has maximum service curve* δ_{τ^b}, *and* N^f *has maximum service curve* δ_{τ^f} *and minimum service curve* δ_T. *The output of* S_1 *is* α_1 *smooth and the second server has strict service curve* β_2. *Denote* $\Delta = T - \tau^f$, *and* $\tau = \tau^f + \tau^b$. *Then the amount of traffic* $B_2(t)$ *in the second*

buffer satisfies

$$B_2(t) \leq E \vee F, \tag{4.12}$$

where

$$\begin{aligned} E &= \vee_{x:0\leq x\leq \tau}\{(\alpha_1 \otimes \overline{W + \delta_\tau})(x+\Delta) - \beta_2(x)\} \\ F &= W - \beta_2(\tau). \end{aligned} \tag{4.13}$$

PROOF. Denote the output from S_1, N^f, S_2 and N^b as R_1, R^f,R_2 and R^b, respectively. W is the window size.

Then the backlog of server 2 at time t is

$$B_2(t) = R^f(t) - R_2(t), \tag{4.14}$$

and the amount of unacknowledged packets at time t is

$$T(t) = R_1(t) - R^b(t) \leq W. \tag{4.15}$$

As N^f, N^b has δ_{τ^f}, δ_{τ^b} as maximum service curve, then

$$\begin{aligned} R^b(t) &\leq R_2 \otimes \delta_{\tau^b}(t) = R_2(t-\tau^b) \\ R^f(t) &\leq R_1 \otimes \delta_{\tau^f}(t) = R_1(t-\tau^f). \end{aligned} \tag{4.16}$$

Therefore,

$$R_1(t) - R_2(t-\tau^b) \leq R_1(t) - R^b(t) \leq W, \tag{4.17}$$

then

$$\begin{aligned} B_2(t) &\leq R_1(t-\tau^f) - R_2(t) \\ &\leq W - (R_2(t) - R_2(t-\tau)). \end{aligned} \tag{4.18}$$

Now define $u = \max\{s : s \leq t, B_2(s) = 0\}$. If $u < t-\tau$, then $B_2(s) > 0$ for all $s \in (t-\tau, t)$ and from the strict service curve property of server 2,

$$R_2(t) - R_2(t-\tau) \geq \beta_2(\tau), \tag{4.19}$$

and

$$B_2(t) \leq W - (R_2(t) - R_2(t-\tau)) \leq W - \beta_2(\tau) = F. \tag{4.20}$$

If $u \geq t-\tau$, then $R^f(u) = R_2(u)$, and $B_2(t) = (R^f(t) - R^f(u)) - (R_2(t) - R_2(u))$.

As N^f has minimum service curve δ_T, and maximum service curve δ_{τ^f}, which means that the delay incurred to each packet is within τ^f and T. Therefore,

$$R^f(t) - R^f(u) \leq R_1(t-\tau^f) - R_1(u-T). \tag{4.21}$$

Furthermore, for any $t - s \leq \tau$, from WFC, we have

$$R_1(t) - R_1(s) \leq W, \tag{4.22}$$

and

$$R_1(t) - R_1(s) \leq \alpha_1(t-s), \tag{4.23}$$

it is easy to see that by (4.21)

$$\begin{aligned} R^f(t) - R^f(u) &\leq R_1(t-\tau^f) - R_1(u-T) \\ &\leq \alpha_1(t-u+\Delta) \wedge (W + \alpha_1(t-u+\Delta-\tau))... \\ &\quad \wedge(nW + \alpha_1(t-u+\Delta-n\tau)). \end{aligned} \tag{4.24}$$

Finally,

$$\begin{aligned} B_2(t) &= (R^f(t) - R^f(u)) - (R_2(t) - R_2(u)) \\ &\leq \alpha_1(t-u+\Delta) \wedge (W + \alpha_1(t-u+\Delta-\tau))... \\ &\quad \wedge(nW + \alpha_1(t-u+\Delta-n\tau)) - \beta_2(t-u) \\ &\leq E. \end{aligned} \tag{4.25}$$

This completes the proof. □

5 Numerical Validations

For numerical validation, we need the expression for the global service curve. Two examples are given to illustrate these calculations, then we derive the conditions for system stability. Next, we introduce the concept of Workload Regulation which allows us to implement the guaranteed service curves needed in simulations. Finally, we report the simulation results for Delay and Backlog bounds and compare them with the corresponding NetCal bounds.

5.1 Global Service Curve and Stability

To ensure that the results of the simulations are useful, we should ensure that the system is stable. The stability condition is described in terms of the global service curve and the arrival curve. So, the global service curve is studied first.

Let us consider a 2-node system consisting of server 1 and server 2. Assume that the size of buffer 1 is infinite and the size of buffer 2 is b (including the server position). The system runs according to the manufacturing blocking mechanism. As shown in Theorem 2.5, the delay of this MB system is dominated from above by a 2-node network under window flow control with the window size $W = b$.

Assume that the isolated server i has service curve β_i $(i = 1, 2)$, then the global service curve of the whole system is $\beta_g = \beta_1 \otimes \beta_2 \otimes \overline{\beta_1 \otimes \beta_2 + W}$, according to Corollary 3.15.

Example 5.1 *Assume that* $\beta_1 = \beta_{r_1,t_1}$, $\beta_2 = \beta_{r_2,t_2}$. *Let* $R = r_1 \wedge r_2$ *and* $T = t_1 + t_2$. *One can prove*

$$\beta_g = \begin{cases} \beta_{R,T}, & \text{if } W > RT, \\ \wedge_{n\geq 0}(\beta_{R,(n+1)T} + nW), & \text{if } W < RT. \end{cases}$$

In Figure 5, we illustrate the global curves β_g for various window sizes W. As the window size W increases, the service curve β_g will increase gradually if $W < RT$ and the service curve β_g remains unchanged regardless of the window size once $W > RT$.

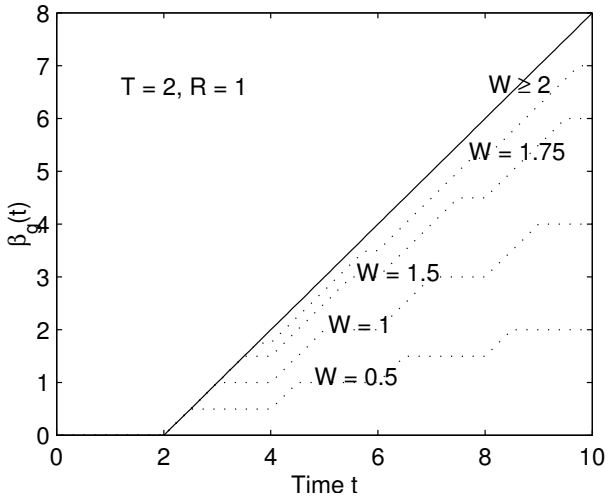
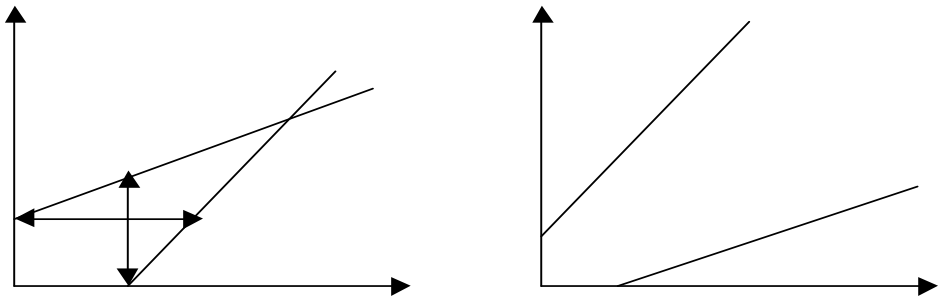


Fig. 5. Global Service Curves of 2-Node System for Different W.

The Figure 5 illustrates that once $W = RT$ (note that W can also be interpreted as the *size* of buffer 2), any further increase of the buffer size does not improve the system service curve. Thus the bounds on delay and backlog obtained by the NetCal technique do not change even if the buffer sizes are further increased. In this sense, the optimal value of buffer size is RT because this finite buffer system has its delay and backlog bounds as small as an infinite buffer system. Note that this is consistent with the result on the optimal buffer size allocation in Subsection 4. Intuitively, this result implies that, when the buffer size is large enough, it is the service at the node instead of the size of the buffer that becomes the limiting factor for the system's global service curve.

After obtaining the global curve β_g, we now consider the stationarity condition of the system. To ensure the arrival stream (from outside network) is constrained by an arrival curve, we require that all arriving data packets pass through a leaky bucket regulator before they enter the network. Hence, the input stream conforms to an affine arrival curve, which is defined as $\alpha(t) = \gamma t + b$. Using $\alpha(t)$ and $\beta_g(t)$, we can obtain a sufficient condition of system stability, which is $\min(W/T, R) \geq \gamma$, or equivalently $\min(W/(t_1 + t_2), r_1, r_2) \geq \gamma$. This

condition can be easily verified by graphing both the curves $\alpha(t)$ and $\beta_g(t)$ on the same axes. In Figure 6, the curve $\beta_g(t)$ is drawn for the case $W > RT$. The following is an intuitive explanation on this condition. To ensure that the delay and backlog bounds are finite, we need the two curves to intersect. It is obvious that the NetCal delay and backlog bounds are infinite for an unstable system. In this case, we say that no NetCal bounds exist.



(a) stable system (b) unstable system

Fig. 6. Explanation on a Condition of System Stability.

Next consider a network with $N > 2$ servers in tandem. Each server except the last one is controlled by the manufacturing blocking mechanism. Let the size of buffer i be b_i, $i = 0, 1, 2, \cdots, N-1$, where $b_0 = \infty$. Then the server i serving the data packets is under a WFC with window size $W_i = b_i$. We know that the global service curve of the whole system has the following form: $\beta_g = (\otimes_{i=1}^{N} \beta_i) \otimes (\otimes_{i=1}^{N-1} \overline{\beta_i \otimes \beta_{i+1} + W_i})$.

Example 5.2 *Consider a system with* 4 *node, and assume that* $\beta_i = \beta_{r_i,t_i}$, $(i = 1, 2, 3, 4)$. *Let* $R = \wedge_{i=1}^{4} r_i$ *and* $T = \sum_{i=1}^{4} t_i$. *Then*

$$\beta_g = \beta_{R,T} \otimes \overline{(\beta_{R_2,T_2} + W_1) \wedge (\beta_{R_3,T_3} + W_2) \wedge (\beta_{R_4,T_4} + W_3)},$$

where $R_i = r_{i-1} \wedge r_i$ and $T_i = t_{i-1} + t_i$ $(i = 2, 3, 4)$.

For ease presentation, we further confine ourselves to the special case, which satisfies (1) $R_2 = R_3 = R_4 = R$ (*e.g.*, $r_1, r_4 \geq r_2 = r_3 = R$); (2) there exists a $k \in \{2, 3, 4\}$ such that $T_k = \max(T_2, T_3, T_4) \equiv T^*$ and $W_k = \min(W_1, W_2, W_3) \equiv W^*$. For this case, we can prove

$$\beta_g = \begin{cases} \beta_{R,T}, & \text{if } W^* > RT^*, \\ \wedge_{n \geq 0}(\beta_{R,T+nT^*} + nW^*), & \text{if } W^* < RT^*. \end{cases}$$

Further, if we assume that this 4 node system has $\alpha(t) = \delta_0 \wedge (at + b)$ as its arrival curve, then a sufficient condition of system stability is $\min(W^*/T^*, R) \geq a$, which can be explained in a manner similar to that in a 2-node system

Note that the NetCal theory offers an elegant characterization of QoS features of queueing systems. However, for applications, we need to regulate arrival and service times that conform with the given arrival and service curves.

The Poisson input process leads to an infinite arrival curve. Consequently, the leaky bucket mechanism has been widely used to regulate the input flows to conform with the given arrival curve (See Figure 4).

We also need to regulate the workload process to offer the guaranteed service curve. So, we introduce the concept of workload regulation. This concept was first introduced in [10] and described as "inverse leaky bucket" therein.

Definition 5.3 (Work Load Regulation) *Let β denote a strict service curve. Define*

$$\gamma(n) = \inf_t \{t : \beta(t) > n - 1\}. \tag{5.1}$$

Notice that γ is essentially the inverse function of the service curve β.

Example 5.4 *If $\beta(t) = R(t-T)^+$, then $\gamma(0) = 0, \gamma(n) = T + (n-1)/R$ has the form of an affine arrival curve, which can be derived from a slotted leaky bucket.*

Denote s_i as the service time of the i-th packet, and $S(n) = \sum_{i=1}^{n} s_i$ the cumulative workload for the first n packets, $S_0 = 0$.

Theorem 5.5 *If the workload process $S(n)$ is regulated by γ defined in* (5.1)*, then the server guarantees the strict service curve β.*

PROOF. To ensure that the server offers strict service curve β, it is sufficient if:

$$\begin{aligned}
s_1 &\le \gamma(1) \\
s_2 &\le \gamma(1) \wedge (\gamma(2) - s_1) \\
&... \\
s_n &\le \gamma(1) \wedge (\gamma(2) - s_{n-1}) \wedge ... \wedge (\gamma(n) - s_{n-1} - s_{n-1} - ... - s_1) \\
&= \gamma(1) \wedge (\gamma(2) - S(n-1) + S(n-2)) \wedge (\gamma(3) - S(n-1) + S(n-3)) \wedge ... \\
&\quad \wedge (\gamma(n) - S(n-1) + S_0) \\
&= (\gamma(1) + S(n-1)) \wedge (\gamma(2) + S(n-2)) \wedge (\gamma(3) + S(n-3)) \wedge ... \\
&\quad \wedge (\gamma(n) + S_0) - S(n-1)
\end{aligned} \tag{5.2}$$

Hence

$$\begin{aligned} & s_n \le \wedge_{i=1}^{n}(\gamma(i) + S(n-i)) - S(n-1) \\ \Longleftrightarrow\ & S(n) \le \wedge_{i=1}^{n}(\gamma(i) + S(n-i)) \\ \Longleftrightarrow\ & S(n) \le \wedge_{i=1}^{n}(\gamma(i) + S(n-i)) \wedge (\gamma(0) + S(n)) \\ \Longleftrightarrow\ & S(n) \le S \otimes \gamma(n). \end{aligned} \tag{5.3}$$

Notice that the last expression gives exactly the same condition as in the definition of arrival curve (Definition 3.4). □

Corollary 5.6 *If the workload process $S(n)$ is regulated by leaky bucket (r, b), then the server guarantees the rate-latency service curve $\beta_{r,(r+b)} = (t - (b + r))^+/r$.*

Remark 5.7 *Theorem 5.5 and Corollary 5.6 suggest a systematic way to modify service times of packets to guarantee the rate-latency service curve $\beta(t) = \beta_{r,(r+b)}$ as follows:*

Pass the workload process $S(n)$ through a slotted leaky bucket (r, b). Denote the output as $S'(n)$, and $s'_i = S'(i) - S'(i-1)$. Then we have

$$S' \le S' \otimes \gamma. \tag{5.4}$$

Therefore, if we use $S'(n)$ as the workload process, i.e., $s'_i = S'(i) - S'(i-1)$ as the service time of the $i-$th packet. Then the system guarantees a strict service curve $\beta_{r,(r+b)}$.

Moreover, if we use multiple leaky buckets instead of single ones, we can create systems with piecewise linear concave (convex) arrival and service curves.

Notice that we can use arbitrary traffic models and workload models for modification. The modified system offers hard QoS bounds, which may be fundamentally different from the characteristics of original queue systems. In this sense, NetCal suggests a different class of traffic/service models to simulate the real system with special emphasis on hard QoS guarantees.

5.3 Simulations for Delay and Backlog Bounds

The NetCal bounds of delay and backlog can be calculated using $\alpha(t)$ and $\beta_g(t)$. To evaluate the tightness of the obtained bounds, we do a simulation study. The Figures 7, 8, 9 and 10 show the simulation results are always bounded by the NetCal bounds, which confirms our theoretical results. The figures 8 and 10 also show that the NetCal backlog bounds can be attained by simulations , which reveals that the NetCal backlog bounds are strictly tight.

Our simulation model is constructed as follows. Packets originate from a Poisson stream with rate λ, which then pass through a leaky bucket (γ, b) before

entering buffer 1. In other words, the arrival process is a leaky-bucket-regulated Poisson stream with arrival curve $\alpha(t) = \gamma t + b$. And the exponentially distributed server (with mean μ_i^{-1}) is treated by the workload regulation technique presented in Section 5.2. Such an approach results in a service node that guarantees the service curve $\beta_i(t) = r_i[t - t_i]^+$. The window size is W, and the number of arrival packets used in our simulation is K. In Figure 7 and Figure 8, the dotted lines are the bounds calculated by the NetCal technique, and the barred lines are simulation results.

Our parameter settings are as follows:

In Figure 7, $\alpha(t) = 0.4t + 1$, $\beta_1(t) = 0.4[t - 10]^+$, $\beta_2(t) = 0.4[t - 10]^+$, $\lambda = 0.5$, $\mu_1 = 0.6$, $\mu_2 = 0.6$, $W = 8$, $K = 1000$.

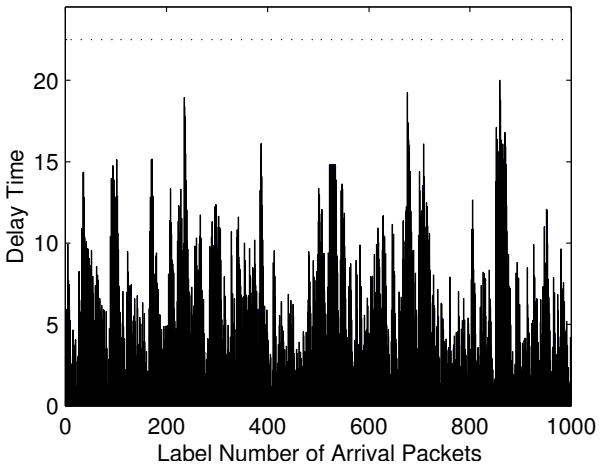


Fig. 7. NetCal Delay Bound of a 2-Node System versus Simulation

In Figure 8, $\alpha(t) = 0.4t + 3$, $\beta_1(t) = 0.4[t - 3]^+$, $\beta_2(t) = 0.5[t - 3]^+$, $\lambda = 0.3$, $\mu_1 = 0.4$, $\mu_2 = 0.5$, $W = 5$, $K = 1000$.

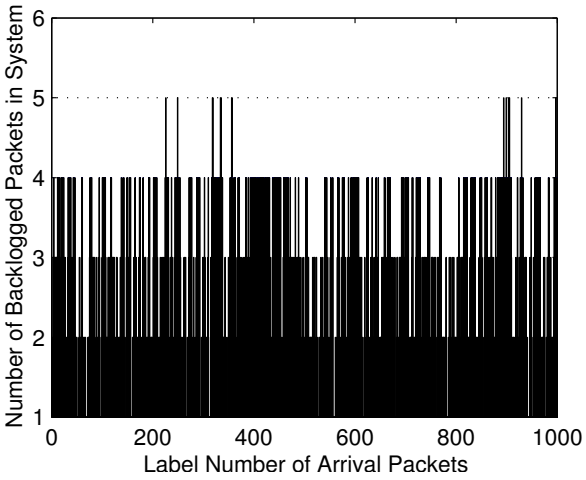


Fig. 8. NetCal Backlog Bound of a 2-Node System versus Simulation.

For the 4-node system we follow the same approach outlined above. As before, the arrival process is a leaky-bucket-regulated Poission stream with arrival curve $\alpha(t) = \gamma t + b$. The service times of packets in server i ($i = 1, 2, 3, 4$) (exponential with mean μ_i^{-1}) are constrained as before to guarantee the service curve $\beta_i(t) = r_i[t - t_i]^+$. The size of buffer i is W_i ($i = 1, 2, 3$), where $W_0 = \infty$. In other words, the window flow control of server i has a window size W_i ($i = 1, 2, 3$). The number of arrival packets used in our simulation is K. In Figures 9 and 10 the dotted lines are the bounds calculated by the NetCal technique, and the barred lines are simulation results. Our parameter settings are as follows: In Figure 9, $\alpha(t) = 0.3t + 1$, $\beta_1(t) = 0.4[t - 2.5]^+$, $\beta_2(t) = 0.3[t - 3.5]^+$, $\beta_3(t) = 0.3[t - 3.5]^+$, $\beta_4(t) = 0.4[t - 2.5]^+$, $\lambda = 0.2$, $\mu_1 = 0.6$, $\mu_2 = 0.7$, $\mu_3 = 0.7$, $\mu_4 = 0.5$, $W_2 = 5$, $W_3 = 4$, $W_4 = 5$, $K = 1000$.

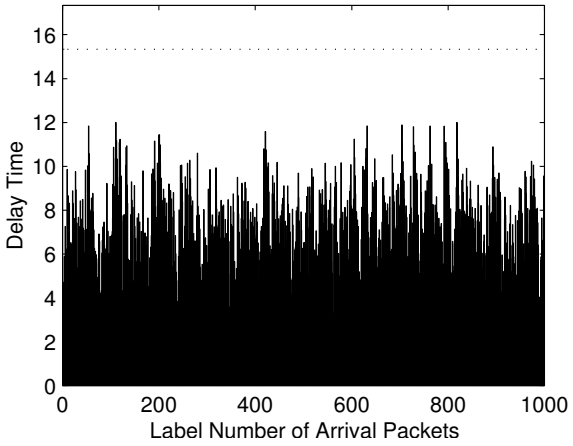


Fig. 9. NetCal Delay Bound of a 4-Node System versus Simulation.

In Figure 10, $\alpha(t) = 0.4t + 1$, $\beta_1(t) = 0.5[t - 3]^+$, $\beta_2(t) = 0.4[t - 3]^+$, $\beta_3(t) = 0.4[t - 3]^+$, $\beta_4(t) = 0.5[t - 3]^+$, $\lambda = 0.3$, $\mu_1 = 0.5$, $\mu_2 = 0.6$, $\mu_3 = 0.5$, $\mu_4 = 0.6$, $W_2 = 4$, $W_3 = 4$, $W_4 = 4$, $K = 1000$.

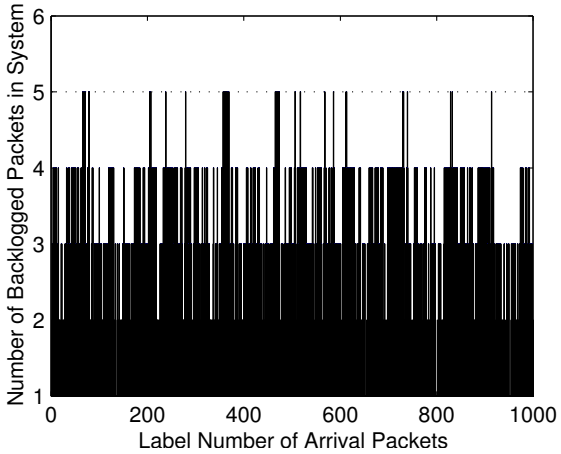


Fig. 10. NetCal Backlog Bound of a 4-Node System versus Simulation.

Remark 5.8 *Further Discussions*

In Figure 7 and Figure 9, an alert reader may notice that the simualtion results of delay time do not hit the delay bound calculated by the NetCal Technique. Now we give an explanation. In Figure 11, we draw a sample path of cumulative numbers of arrival/departures, where the dotted lines represent the simulated delay times, and bold line represents the NetCal bound of delay time. One can easily find that the simulated delay time can not reach the NetCal delay bound except for some extremely rare cases. The reason is that arrival and service processes are constrained by the affine curve $\alpha(t) = \gamma t + b$ and the rate latency curve $\beta(t) = r[t - c]^+$, respectively. In this case, NetCal delay bound equals the distance between the points $(0, b)$ and $(c + r^{-1}b, b)$. Generally speaking, the simulated arrival and departure sample paths (the staircase lines) can not simultaneously reach the point $(0, b)$ and $(c+r^{-1}b, b)$, respectively. Intuitively, more bursty traffic (than the Poisson processes) might be able to result in a higher worst-case delay. Finally, the delay bound in this example is obtained from WFC system directly. From the dominance relationship between WFC and MB, it is clear that for the MB systems, the delay bound is harder to achieve.

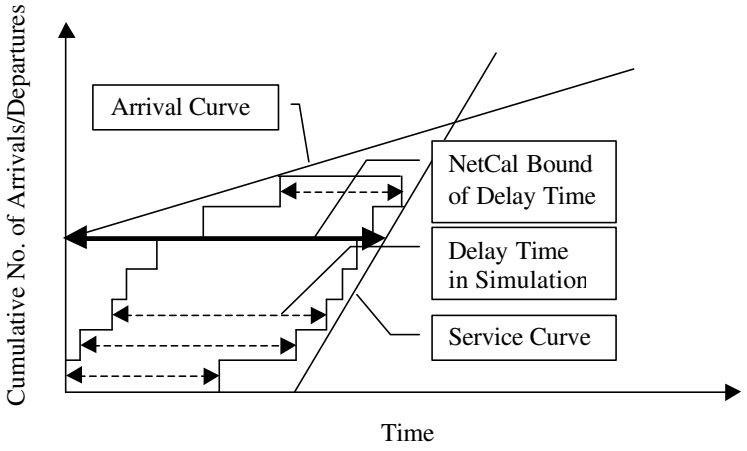


Fig. 11. Explanation of Simulation Results that Can Not Hit the NetCal Delay Bound.

Acknowledgements

We would like to thank Peter Rabinovitch of Alcatel for posing the problem discussed in this paper and many fruitful discussions.

References

[1] A. Brandwajn, and Y. L. Jow, An approximation method for tandem queues

with blocking, *Operation Research* , **36** (1988) 73-83.

[2] J.A. Buzacott, and J.G. Shanthikumar, *Stochastic Models of Manufacturing Systems.* Prentice Hall, 1992.

[3] C. S. Chang, On deterministic traffic regulation and service guarantees: a systematic approach by filtering, *IEEE Transactions on Information Theory*, **44** (1998) 1097-1110.

[4] C. S. Chang, Matrix extensions of the filtering theory for deterministic traffic regulation and service guarantees, *IEEE J. Selected Areas in Communications*, **16** (1998) 708-718.

[5] C. S. Chang, *Performance Guarantees in Communication Networks*, Springer, 2000.

[6] R.L. Cruz, A Calculus for Network Delay, Part I: Network elements in isolation, *IEEE Tran. Inform. Theory,* **37** (1991) 114-131.

[7] R.L. Cruz, A Calculus for Network Delay, Part II: Network elements in isolation, *IEEE Tran. Inform. Theory,* **37** (1991) 132-141.

[8] R.L. Cruz, and C.M. Okino, Service Gurantees for Window Flow Control. in: *Proceddings 34 Allerton Conference on Communication*, Control & Computing. Monticello, IL, Oct. (1996)

[9] D. P. Gaver, P. A. Jacobs, and G. Latouche, Finite birth-and-death models in randomly changing environments. *Advances in Applied Probability*, **16** (1984) 715-731.

[10] X. Jiang, *Performance Analysis with Network Calculus Approach, M. Eng Dissertation, University of Toronto*, 2002.

[11] S. Khorsandi, and A. Leon-Garcia, Robust non-probabilistic bounds for delay and throughput in credit-based flow control. in: *Proceedings IEEE INFORCOM'96* (1996) **2**, pp. 577-584.

[12] J.Y. Le Boudec, Application of network calculus to guarantee service network, *IEEE Transactions on Information Theory*, **44** (1998) 1087-1096 .

[13] J.Y. Le Boudec, and P. Thiran, *Network Calculus - A Theory of Deterministic Queuing Systems for the Internet*, L.N.C.S #2050, Springer, 2001.

[14] I. Mitrani, *Probabilistic Modelling*, Cambridge University Press, Cambridge, 1998.