Monotonicity of the throughput of a closed queueing network in the number of jobs
Adan, I.J.B.F.; van der Wal, J.

Published: 01/01/1987

Document Version
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the author's version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

Citation for published version (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal?

Take down policy
If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.
Memorandum COSOR 87-03

Monotonicity of the throughput of a closed queueing network in the number of jobs

by

Ivo Adan and Jan van der Wal

Eindhoven, The Netherlands

February 1987
Monotonicity of the throughput of a closed queueing network in the number of jobs.

by

Ivo Adan and Jan van der Wal
University of Technology, Eindhoven

ABSTRACT

Using a sample path argument it is shown that the throughput of a closed queueing network with general service times is nondecreasing in the number of jobs.

1. Introduction.

The last two years various authors have addressed the problem of proving that the throughput in a closed queueing network is nondecreasing in the number of jobs in the network.

For product form networks several proofs have been given, see e.g. Robertazzi and Lazar [1985], Suri [1985], Yao [1985], Van der Wal [1985] and Shanthikumar and Yao [1985]. For non-product form networks only partial results exist. E.g. in a previous paper we established the monotonicity for the case of Erlang service times, see Adan and Van der Wal [1987]. It is likely that this proof can be extended to the case of phase-type service times. In a somewhat different way monotonicity is studied to obtain estimates for the performance of non-product form networks by bounding the network between product form networks, see e.g. Van Dijk and Lamond [1986].

In this paper we use a sample path argument to establish the intuitively obvious result that for the closed queueing network with general service times the throughput does not decrease if an extra job is added to the network.

The paper is organized as follows. In section 2 the model, some notations and the theorem are given. In section 3 the theorem is proved for the case that all stations are single servers. The case of multi servers is treated in section 4.


Consider a closed queueing network with stations 1, 2, ..., N and general, though independent service times. In station i there are L_i identical servers (possibly L_i = ∞), i = 1, ..., N and in each station the discipline is FCFS.

In order to establish the monotonicity we shall compare the queueing networks with K and K+1 jobs. We shall show that, for two specific initial states and a given realization of the sequences of service times in the various queues and the transitions to be made, the throughput up to time t in the K+1 job system is at least
equal to the one in the K job system.

Let $X_{ij}$ and $S_{ij}$, $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots$ be any given realization of service times and transitions. I.e. $X_{ij}$ is the service time required by the $j$-th arriving job in queue $i$ and $S_{ij}$ be the station the $j$-th departing job from queue $i$ will jump to after his service is completed. In the multi server case the $j$-th departing job need not be the $j$-th arriving job. For the time being one may think of the $S_{ij}$'s as outcomes of a Markovian routing.

In both the K and K+1 job system we assume all jobs to arrive at $t = 0$ in queue 1. This assumption is convenient but will be relaxed later on.

We need some further notations.

- $A_{ij}$: the time of the $j$-th arrival in queue $i$.
- $D_{ij}$: the departure time of the job that arrived as $j$-th job in queue $i$. In the multi server case this need not be the time of the $j$-th departure.
- $A_i(t)$: total number of arrivals up to and including time $t$ in queue $i$.
- $D_i(t)$: total number of departures up to and including time $t$ from queue $i$.

In the sequel these variables will have a superscript $K$ or $K+1$ to indicate whether they correspond to the K or the K+1 job network.

It will be clear that in order to prove the monotonicity it is sufficient to show that for any realization of $X_{ij}$'s and $S_{ij}$'s we have for all $i$ and $t$

$$D_{i, K+1}(t) \geq D_{i, K}(t).$$

I.e. for any realization it is rewarding to have an extra job in the system.

Finally let $a^C_1 < a^C_2 < \ldots$ be the time instants in the C job system upon which one or more services are completed. Then define the sequence $t_0, t_1, \ldots$ by

$$t_0 := 0$$
$$t_n := \min \{ \min \{ a_i^K \mid a_i^K > t_{n-1} \}, \min \{ a_i^{K+1} \mid a_i^{K+1} > t_{n-1} \} \}, \ n \geq 1.$$

So $\{t_n\}$ is the sequence of time instants upon which something happens in at least one of the two systems.

We make the following assumption

**Assumption**

(i) $X_{ij} > 0$ for all $i$ and $j$

(ii) $\sum_{j=1}^{\infty} X_{ij} \to \infty \ (n \to \infty)$ for all $i$

The condition $X_{ij} > 0$ guarantees that a job can complete only one service at a time and hence make at most one transition at a time. The second condition guarantees $t_n \to \infty$ for $n \to \infty$ (see Appendix).
Now we can state our main result that for all \( t \) and for each station the throughput in the \( K+1 \) system is at least equal to the one in the \( K \) job system. Recall that all jobs arrive in queue 1 at \( t = 0 \), so \( A_i^K(0) = C \) and \( A_i^C = 0 \), i \( \neq 1 \), \( C = K, K+1 \), and since \( X_{ij} > 0 \) for all \( i \) and \( j \) also \( D_i^C = 0 \) for all \( i \) and \( C = K, K+1 \).

**Theorem.**

For all \( t \geq 0 \) and all \( i = 1, 2, ..., N \)
\[
(1) \quad D_i^{K+1}(t) \geq D_i^K(t)
\]

Since the functions \( D_i^C \) are constant on the intervals \([t_m, t_{m+1})\) and \( t_n \to \infty \) \((n \to \infty)\) it suffices to prove the theorem for the instants \( t_0, t_1, ... \).

3. The single server case.

The proof will be based on the following rather trivial but vital lemma stating that if arrivals come earlier then so do departures.

**Lemma 1. (Single server)**

If station \( i \) is a single server and
\[
A_{ij}^{K+1} \leq A_{ij}^K \quad \text{for} \quad j = 1, 2, ..., n
\]
then
\[
D_{ij}^{K+1} \leq D_{ij}^K \quad \text{for} \quad j = 1, 2, ..., n
\]

**Proof.**

By induction.

\( n = 1 \).

\[
D_{1}^{K+1} = A_{11}^{K+1} + X_{11} \leq A_{11}^K + X_{11} = D_{1}^K
\]

Assume that the lemma holds for \( n = m \). Then
\[
(2) \quad D_{im+1}^{K+1} = \max \{ D_{im}^{K+1}, A_{im+1}^K \} + X_{im+1}
\]
\[
\leq \max \{ D_{im}^K, A_{im+1}^K \} + X_{im+1} = D_{im+1}^K
\]
which proves the lemma for \( n = m+1 \). \( \square \)

As argued before it suffices to prove (1) for the sequence \( t_0, t_1, ... \). This will be done by induction.

For \( t_0 \) the inequality (1) holds by definition (all \( D_i^C(0) = 0 \)).

Assume (1) holds for \( t_0, t_1, ..., t_m \). Then, with \( \delta(i, j) = 1 \) if \( i = j \) and 0 otherwise, for \( k = 0, 1, ..., m \)
\[
A_i^{K+1}(t_k) = \sum_{l=1}^{N} \sum_{j=1}^{D_{ij}^{K+1}(t_k)} \delta(S_{lj}, i) + (K+1) \delta(i, 1)
\]
\[
\geq \sum_{l=1}^{N} \sum_{j=1}^{D_{ij}(t_k)} \delta(S_{lj}, i) + K \delta(i, 1)
\]
= A_i^K(t_k).

Since \( A_i^C(t) \) is constant on \([ t_{k-1} , t_k )\)
\( A_i^{K+1}(t) \geq A_i^K(t) \) for \( t \leq t_m \)
thus
\[ A_{ij}^{K+1} \leq A_{ij}^K \] for \( j = 1 , \ldots , A_i^K(t_m) \).

and by lemma 1
\[ D_{ij}^{K+1} \leq D_{ij}^K \] for \( j = 1 , \ldots , A_i^K(t_m) \).
Since \( X_{ij} > 0 \)
\[ D_i^K(t_{m+1}) \leq A_i^K(t_m) . \]
thus
\[ D_{ij}^{K+1} \leq D_{ij}^K \] for \( j = 1 , \ldots , D_i^K(t_{m+1}) \).
Hence
\[ D_i^{K+1}(t_{m+1}) \geq D_i^K(t_{m+1}) . \]
So (1) holds for \( t_{m+1} \).
This completes the proof of the theorem for the single server case.

4. The multi server case.
The proof for the multi server case follows exactly the same lines as the single server one. Once the multi server equivalent of lemma 1 is established the rest of the argument for the proof of the theorem is identical.
The problem is to prove lemma 1 for multi servers. For that we need more notation.
Define for \( C = K \cdot K+1 \)
\[ Y_{ij}^C = \begin{cases} 1 & \text{if the } j-\text{th arriving job in station } i \\
& \text{is served by server } l \\
0 & \text{otherwise} \end{cases} \]
\[ T_{ij}^C \] the time server \( l \) in station \( i \) becomes available for the \( j-\text{th arriving job} \), so
\[ T_{ij}^C = \max \{ D_{ik}^C Y_{ik}^C , k = 1 , 2 , \ldots , j-1 \} \]
and
\[ T_{ij}^C \] the vector of \( T_{ij}^C \)'s
\[ T_{ij}^C = ( T_{ij}^C , T_{ij}^C , \ldots ) \]
Now we have
\[ D_{ij}^C = \max \{ \min_k T_{ij}^C , A_{ij}^C \} + X_{ij} \]
which clearly is more complicated than the relation for $D_{ij}^C$ in (2). So in order to establish the multi server equivalent of lemma 1 we have to study the $T_{ij}^C$'s. The case $L_i = \infty$ is simple since then (3) reduces to

$$D_{ij}^C = A_{ij}^C + X_{ij}. \tag{4}$$

so $A_{ij}^{K+1} \leq A_{ij}^K$ implies that $D_{ij}^{K+1} \leq D_{ij}^K$. Now assume $L_i < \infty$.

$$T_{ij}^C = \begin{cases} \max \{ T_{ij}^C, A_{ij}^C \} + X_{ij}, & \text{if } T_{ij}^C = \min_k T_{ik}^C \\ \text{and (the tiebreak rule) } T_{jk}^C > T_{ij}^C, & k < l \\ T_{ij}^C, & \text{otherwise} \end{cases} \tag{5}$$

So $T_{ij}^{K+1}$ is obtained from $T_{ij}^C$ by replacing the (a) minimal component. To study the effect of this we need the following.

Let $a = (a_1, a_2, ..., a_n)$ be a vector. Then $Ua$ denotes the nondecreasing reordering of $a$:

$$Ua = (a_{i_1}, a_{i_2}, ..., a_{i_n})$$

with $a_1 \leq a_2 \leq ... \leq a_n$, and $(i_1, i_2, ..., i_n)$ a permutation of $(1, 2, ..., n)$. For two vectors $a = (a_1, a_2, ..., a_n)$ and $b = (b_1, b_2, ..., b_n)$ we write

$$a \preceq b \text{ if } (Ua)_i \leq (Ub)_i, i = 1, ..., n.$$ 

One may easily verify the following result.

**Lemma 2.**

Let

$$(a_1, a_2, ..., a_n) \preceq (b_1, b_2, ..., b_n), \alpha \leq \beta, a_i = \min_l a_j, b_j = \min_l b_j$$

then

$$(a_1, a_2, ..., a_{i-1}, \alpha, a_{i+1}, ..., a_n) \preceq (b_1, ..., b_{j-1}, \beta, b_{j+1}, ..., b_n).$$

So replacing a minimal component in $a$ by $\alpha$ and in $b$ by $\beta$ with $\alpha \leq \beta$ does not affect the ordering $a \preceq b$.

This gives us

**Corollary.**

If

$$T_{ij}^{K+1} \preceq T_{ij}^K \text{ and } A_{ij}^{K+1} \leq A_{ij}^K$$

then

$$T_{ij}^{K+1} \preceq T_{ij}^K.$$
Proof.
Immediate from (5) and lemma 2.

Now we can state and prove the multi server version of lemma 1.

**Lemma 3. (Multi server)**

If station $i$ is a multi server and

$$A_{ij}^{K+1} \leq A_{ij}^K \text{ for } j = 1, 2, \ldots, n$$

then

$$D_{ij}^{K+1} \leq D_{ij}^K \text{ for } j = 1, 2, \ldots, n$$

**Proof.**

As observed before we only need to consider the case $L_i < \infty$ as the lemma is trivially obtained from (4) for $L_i = \infty$. Since $T_i^c = (0, \ldots, 0)$, so $T_{ij}^{K+1} = T_{ij}^K$ (guaranteeing the initial condition of the corollary) the corollary yields us

$$T_{ij}^{K+1} \leq T_{ij}^K$$

and thus

$$\min_i T_{ij}^{K+1} \leq \min_i T_{ij}^K \text{ for } j = 1, 2, \ldots, n.$$ 

By (3) this implies

$$D_{ij}^{K+1} \leq D_{ij}^K \text{ for } j = 1, 2, \ldots, n,$$

which completes the proof of the lemma.

As said before, this gives the proof of the theorem for multi servers.

5. Conclusions and remarks.

From the preceding we conclude that for a closed queueing network with in each station nonzero, independent and identically distributed service times with Markovian routing the throughput upto time $t$, defined as the expected number of service completions in a station upto and including time $t$ is nondecreasing in the number of jobs in the system for all $t$, if initially all jobs are in one queue. So also the average number of service completions is nondecreasing.

In the remainder of this section we relax most of the above conditions, such as i.i.d. service times, Markovian routing and all jobs initially in one queue.

**I.i.d. service times.**

- The successive service times in a station may be dependent as long as the $X_{ij}$ are independent of the $A_{ij}, X_{kl}, k \neq i, \text{ etc.}$ Usually dependence between the successive $X_{ij}$ is caused by something like the temporarily malfunctioning of a server. In that case the dependence between $X_{ij}$ and $X_{ij+1}$ is stronger if the times at which the $j$-th and the $j+1$-th job start their service are closer, so there also is a dependence between $\{X_{ij}\}$ and $\{A_{ij}\}$. Then the coupling of service times as used here is no longer possible. I.e. we can no longer say that the $X_{ij}$ in the $K$ and $K+1$ job systems are...
Another type of dependency in service times might be more natural. The service times of a specific job out of the K or K+1 jobs are dependent, e.g. (nearly) the same in each station. In this case monotonicity is no longer guaranteed. For example, consider a system consisting of two single server stations, where after each service completion the jobs move to the other station. The service times of a job are equal in both stations and at all visits. Initially for each job \( k \) a service time \( x_k^1 \), \( k = 1, \ldots, K+1 \) is drawn from a discrete distribution, with \( P(x_k^1 = 1) = 1/5 \) and \( P(x_k^2 = 100) = 4/5 \), and the job will keep this service time for the rest of its life. The \( x_k^1 \), \( k = 1, \ldots, K+1 \) are independent. Then we find for the throughput \( T(K) \) defined as the average number of service completions in the two stations together:

\[
T(1) = \frac{1}{5} \cdot 1 + \frac{4}{5} \cdot \frac{1}{100} = .0208
\]

and

\[
T(2) = \frac{1}{25} \cdot 2 + \frac{24}{25} \cdot \frac{2}{100} = .0992 < T(1)
\]

Also the service times need not be identically distributed. For instance every twentieth job in a station may have a 50% larger service time due to, say, monitoring of the servicing or inspection.

Markovian routing.

- Any routing as long as it can be characterized by a sequence of random variables \( S_{ij} \), \( j = 1, 2, \ldots \) independent of everything else going on in the network is allowed. We only need that the \( S_{ij} \) in the K and K+1 job system can be taken to be the same, by a coupling argument. For instance alternating routing is possible: \( S_{ij} = 1 \) if \( j \) is even and \( S_{ij} = 2 \) if \( j \) is odd.

- In the single server system the \( S_{ij} \) and \( X_{ij} \) may be dependent. For the multi server case the coupling fails because the j-th departing job is not necessarily the one receiving time \( X_{ij} \).

All jobs initially in one station.

- It is clear that this can be relaxed to: the number of jobs initially in station \( i \) in the K+1 job system is at least equal to the number in the K system for \( i = 1, \ldots, N \). Furthermore, under very weak conditions the average number of service completions in a station per unit of time is independent of the initial state.
Appendix.

If

\[ \sum_{j=1}^{n} X_{ij} \rightarrow \infty \text{ for } n \rightarrow \infty \]

then

\[ t_n \rightarrow \infty \]

Suppose to the contrary \( t_n \rightarrow t < \infty \) for \( n \rightarrow \infty \), then \( a_n \rightarrow t (n \rightarrow \infty) \) for \( C = K \) or \( K+1 \), say \( C = K \). Then we have in at least one station, say \( i \),

\[ A_i(x) \rightarrow \infty \quad (x \rightarrow t) \]

Let us mark the \( K \) jobs in the system 1, 2, ..., \( K \), and let \( J_1 \) be the subset of \{1, 2, ..., \( K \)\} of jobs which arrive infinitely often in station \( i \) in \([0, t)\). So \( J_1 \) is nonempty. Let \( J_2 \) be the complement: \( J_2 = \{1, ..., \( K \}\} \setminus J_1 \).

Define for each job \( k \in J_2 \)

\[ t_k^A \text{ the last arrival in station } i \text{ before } t \]

\[ t_k^D \text{ the last departure station } i \text{ before } t \]

and define

\[ t_0 = \max \max_{k \in J_2} (t_k^A \cdot t_k^D) \]

Then each job \( k \in J_2 \) is either stuck in a task \( X_{ij} \) during the whole interval \((t_0, t)\), if \( t_k^A > t_k^D \). or it is not in station \( i \) for all \( x \) in \((t_0, t)\), if \( t_k^D > t_k^A \).

So there is a finite, possibly empty set \( I \) of indices of tasks performed by the \( J_2 \) jobs between \( t_0 \) and \( t \). Since \( I \) is fixed and finite and since \( A_i(x) \rightarrow \infty \) for \( x \rightarrow t \) and all \( X_{ij}, j \notin I \), are completed before \( t \)

\[ \sum_{j=1}^{n} A_i(x) X_{ij} \rightarrow \infty \text{ if } x \rightarrow t. \]

On the other hand each job can spend at most \( x \) time units in a specific station between \( 0 \) and \( x \), hence for all \( x < t \)

\[ \sum_{j=1}^{n} X_{ij} \leq Kt < \infty \]

Contradiction, so \( t_n \rightarrow \infty \) for \( n \rightarrow \infty \).
References.


