

# Scalability of Fork/Join Queueing Networks with Blocking

Cathy H. Xia, Zhen Liu  
IBM T.J. Watson Research  
Center  
Hawthorne, NY 10532, USA  
cathyx@us.ibm.com,  
zhenl@us.ibm.com

Don Towsley\*  
Dept. of Computer Science  
University of Massachusetts  
Amherst, MA, USA  
towsley@cs.umass.edu

Marc Lelarge  
INRIA & Ecole Normale  
Supérieure  
75005 Paris, France  
Marc.Lelarge@ens.fr

## ABSTRACT

This paper investigates how the throughput of a general fork/join queueing network with blocking behaves as the number of nodes increases to infinity while the processing speed and buffer space of each node stay unchanged. The problem is motivated by applications arising from distributed systems and computer networks. One example is large-scale distributed stream processing systems where TCP is used as the transport protocol for data transfer in between processing components. Other examples include reliable multicast in overlay networks, and reliable data transfer in ad hoc networks. Using an analytical approach, the paper establishes bounds on the asymptotic throughput of such a network. For a subclass of networks which are balanced, we obtain sufficient conditions under which the network stays scalable in the sense that the throughput is lower bounded by a positive constant as the network size increases. Necessary conditions of throughput scalability are derived for general networks. The special class of series-parallel networks is then studied in greater detail, where the asymptotic behavior of the throughput is characterized.

## Categories and Subject Descriptors

C.4 [Computer Systems Organization]: PERFORMANCE OF SYSTEMS

## General Terms

Theory, Performance

## Keywords

Fork and Join, Blocking, Queueing Networks, Throughput, Scalability, Asymptotic Analysis

---

\*This work was supported in part by the National Science Foundation under grant EEC-0313747.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SIGMETRICS'07, June 12–16, 2007, San Diego, California, USA.  
Copyright 2007 ACM 978-1-59593-639-4/07/0006 ...\$5.00.

## 1. INTRODUCTION

With the rapid advances in computer technology and wireless communications, distributed stream processing has emerged as an appealing solution for analyzing large amounts of data from dispersed sources. In such a paradigm, massive numbers of real-time streams are collected, processed, and aggregated across a large number of interconnected devices. The scalability of such large-scale networks is of critical importance, which has been widely recognized by practitioners. Providing a framework to assess the conditions under which the throughputs of such networks do or do not scale is urgently needed to enable the efficient design of such networks in large scale.

The design of scalable stream processing systems has received considerable attention in the past five years, see, for example, Borealis [1], Medusa [12], GATES [11], IrisNet [19] and SPC [22]. While this design problem has been addressed by practitioners and system engineers, there has been little effort to establish a mathematical foundation for scalable stream processing. Since building experimental large-scale networks is very expensive and practically infeasible, a theoretical approach therefore has a unique advantage in providing architects and engineers qualitative insights about the scaling properties of various network structures.

We consider the case where stream processing networks deploy reliable transport protocols such as TCP for data transfer between nodes, as is the case in the SPC system [22], as well as in increasingly more streaming applications [20, 28, 33, 29]. Further benefits of using TCP instead of UDP for such applications include fair bandwidth sharing, in-order delivery, and local recovery from packets losses.

Such TCP-type congestion control between nodes has also been advocated for improving the reliability of end-system multicast in one-to-many overlay networks [2, 10, 23, 25], and for supporting multimedia streaming in mobile ad hoc networks [17, 21]. In all these cases, TCP implements back pressure mechanism between neighboring buffers to prevent overflow. The scalability of such lossless networks has received increasing attention [2, 16, 26].

The performance of the above lossless networks is limited by multiple factors, including processing speed, buffering/storage capacity, and the coordination of input and output between various processing nodes. Without careful coordination, a full buffer can block upstream processing and result in poor overall system throughput. Such inefficiency may be tolerable in a small network, but can accumulate and become dominant in determining the performance of a large network. In this regard, we investigate the behavior

of a lossless network as its size grows while the processing speed and buffer space of each device remain constant. We say that a network architecture is scalable, if its throughput does not go to zero as its size grows, when the processing speed and buffer space in each node are fixed. Such scalability is also of particular importance in wireless sensor networks, where devices typically have limited resources.

We are interested in the throughput scalability of large stream processing systems such as SPC [22] with TCP-type reliable data transfer. We model such a stream processing system as a queueing network with fork/join mechanisms and finite-capacity buffers, which we will refer to as Fork/Join Queueing Networks with Blocking (FJQN/Bs). Such FJQN/B models capture three key aspects common to many practical stream processing applications: the simultaneous consumption of flows (join), multiple outputs (fork), and blocking. A variety of processing functions in streaming applications may *simultaneously* require multiple input data streams and produce one or more valuable output streams. Operations such as joins and aggregates are the core of most stream processing [15]. For example, in the surveillance field, one may require the synchronized processing of audio and video data in order to generate accurate, up-to-date information. Multiple output streams may also be generated in this case, as relevant segments of audio and video data may be retained and passed to downstream processing nodes for further analysis. Such a network can be generally described by a directed graph, where vertices represent the processing nodes and directed edges represent information flows. A vertex with several incoming edges corresponds to an operation that requires synchronization (or join). A vertex with several outgoing edges corresponds to forking of the output. Such a representation is similar to that of a dataflow network [30]. Blocking of service occurs when a buffer reaches full capacity.

Note that TCP uses a variable congestion window size for each pair of sender and receiver. This type of window control is not easily modeled by a queueing network with finite buffer. However, thanks to the monotonicity property (see [3, 6, 2]), the throughput of the TCP session is upper and lower bounded by the cases of fixed window sizes with the maximum and minimum windows, respectively. These networks with fixed feedback window control mechanism can be modeled by the queueing networks with blocking, see more detailed discussions in the next section.

We focus on the throughput scalability of the above FJQN/B networks as the networks grow in size. Our problem can be considered as the *throughput scalability* problem for general FJQN/Bs, which can also be cast into the various settings of TCP-controlled one-to-many overlay networks and mobile ad hoc networks as mentioned earlier, as well as broad applications including parallel programming, manufacturing, supply chain management, or other microeconomic practice, see e.g. [5, 27, 31].

Most previous works on FJQN/Bs have focused on performance properties for a given FJQN/B network. For example, [4] studied the stability condition of such networks, [13] established duality, reversibility, symmetry properties, [6] and [14] derived various stochastic comparison results. Scaling properties of various networks have received increasing attention in the recent years. It was shown in [7, 9] that for IP-supported reliable multicast schemes using a TCP like control, the group throughput decreases and tends to zero

when the group size increases. Negative results are also reported in [24] in the context of lossy wireless networks where it was shown that throughput reduces to zero in a lossy finite buffer tandem network as the number of nodes increases to infinity. FJQN/B networks can be considered as lossless networks where service is blocked when a buffer reaches full capacity. [26] shows that, for a tandem network with blocking, throughput is lower bounded by a positive constant independent of the network size. Using (max,plus)-algebraic simulation, [2] also reports positive results in a tree network in the context of one-to-many TCP overlay networks.

Our contributions are three-fold.

- We provide a mathematical framework for analyzing the throughput of general FJQN/B's under arbitrary network topologies. We provide upper and lower bounds on the asymptotic throughput as the network size grows. We further derive necessary conditions for a FJQN/B to be throughput scalable, i.e., for its throughput to remain bounded away from zero as the network size increases. In general, throughput scalability requires the FJQN/B network to have bounded node-degree and to satisfy a certain “balance” condition. Such balance requires the ratio of the number of nodes/buffers on two paths between any pairs of nodes to remain within a certain ratio.
- We study extensively the class of series-parallel networks, which is a special class of FJQN/Bs. We provide explicit conditions under which the throughput scales and conditions under which it does not. In cases that throughput does not scale, we further conduct asymptotic analysis and derive the exact rate of the throughput degradation.
- We report positive results for a large class of FJQN/B networks, namely those that are “balanced” and not too “wide”. Briefly balance refers to the relative lengths of paths between pairs of nodes and width of a FJQN/B refers to the maximum number of nodes that do not have paths between them. We show that the throughput of these networks remains bounded away from zero regardless of their size.

Our results can also be applied to further the understanding of the asymptotic behavior of general queueing networks with finite buffers in large scale. For example we can show that, using the duality property, our result in a two-branch series-parallel network can be interpreted in the closed queueing network setting with finite buffers. Notice that for such a general queueing network with finite buffers, there are no explicit solutions such as the product-form of Jackson networks.

The paper is organized as follows. The next section describes the model and the mathematical framework under which the throughput of a FJQN/B network is defined. Preliminary results on performance properties of FJQN/B's are presented in Section 3. In Section 4, various upper and lower bounds on throughput are established for general FJQN/B's, and we derive necessary conditions for the system to be throughput scalable. Detailed analysis for series-parallel networks is presented in Section 5. In Section 6, we establish positive scalability results for a large class of  $w$ -wide FJQN/B networks. Concluding remarks are given in Section 7.

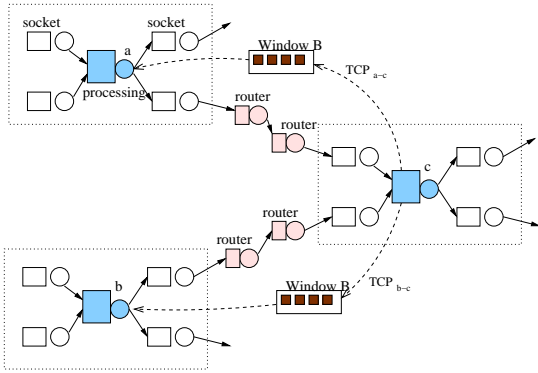


Figure 1: Example: Stream Processing Network

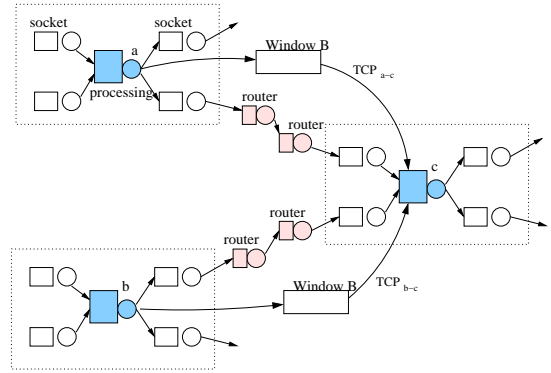


Figure 2: Dual of the Above Network.

## 2. MODEL

### 2.1 System Model Descriptions

We consider a distributed data processing system (e.g. as illustrated in Figure 1), consisting of  $N$  nodes connected to each other by a set of (possibly logical) communication links. We refer to such a system as stream processing network. For the sake of discussions we shall assume that the network is connected, directed and acyclic. The reader will see that some of the results do extend to the case of cyclic networks. The nodes without incoming links are called source nodes, whereas those without outgoing links are called sink nodes. Data are generated by source nodes and consumed by sink nodes after being processed in the network. When a node has multiple incoming links, the node needs to process a data item from each of these links in a synchronous way, i.e. there should be one data item available from each of these links in order to proceed with the analysis/computation task of that node. Similarly, when a node has multiple outgoing links, then a data item is sent out to each of these links. These data items need not be identical though.

Note that in a more general framework, nodes can process the data items from incoming links in an asynchronous way or in a partially synchronous way (with only a subset of links to be simultaneously processed). Similarly, the production of data items feeding downstream nodes can be performed in an asynchronous way, i.e., data items are sent only to a subset of outgoing links. As our main interest is in the scalability of such systems, we shall only consider the worst case, where the consumption and the production are both synchronous. Such mechanisms are also called *fork* (simultaneous production) and *join* (simultaneous consumption) operations in the literature.

We assume the network uses a reliable transport protocol such as TCP for data transfer between nodes, as illustrated in Figure 1. We shall use queueing networks with blocking to represent the feedback control on the communication link and back-pressure in the nodes. It is worthwhile noting that TCP uses a variable window size which is not easily modeled by a fixed-size finite buffer. However, the finite buffer with minimum (resp. maximum) window size is the worst (resp. optimistic) scenario in terms of the throughput of the TCP session. Thus, the scalability (or non-scalability) result pertaining to the lossless case with fixed buffer size can be extended to the general case when TCP is used for the data transfers.

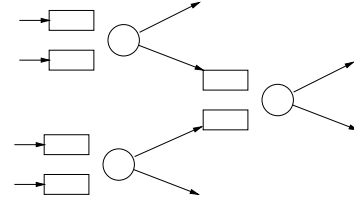


Figure 3: Simplified network of that in Figure 1.

Consider Figure 1 for example, TCP control mechanism is used for the data transfer between nodes  $a$  and  $c$ , and between nodes  $b$  and  $c$ . We first replace the (generally variable in size) TCP window with a buffer of fixed size (e.g. minimum window size, say  $B \geq 1$ ) and assume it is initially full. We then convert the system into its dual system obtained by reversing the flow direction, and replacing the jobs in the buffer (TCP window) with holes. See Figure 2. Based on the duality of [13], the dual system is stochastically equivalent to the original network.

For the case of throughput lower bound, typically the minimum window size of TCP is 1. In this case, the network of Figure 2 can further be simplified to that of Figure 3, due to the fact that there is at most one packet in-between nodes  $a$  and  $c$ , and in-between nodes  $b$  and  $c$ . Similarly, for the throughput upper bound case, we can consider the case where routers are not the bottleneck, so that the two parallel paths in-between nodes  $a$  and  $c$  can be reduced to one path with the buffer size to be equal to the smaller of the two. The same argument holds for the paths in-between nodes  $b$  and  $c$ . Thus, again, from topological perspective, the network of Figure 3 can be used for the derivation throughput upper bounds, which will in turn allow us to obtain necessary conditions for a system to have scalable throughput.

### 2.2 Fork/Join Queueing Network with Blocking

In what follows we model a stream processing network as a fork/join queueing network with blocking (FJQN/B). Let

$$\mathcal{N} = (V, E, \mathbf{B})$$

be an acyclic directed graph where  $V$  is a set of  $n$  servers,  $E \subset V^2$  is a set of directed edges indicating the flow of jobs from servers to servers. Associated with edge  $(i, j) \in E$  is a buffer with finite capacity  $B_{i,j} \in \mathbb{N}$ . It is convenient to refer

to the underlying graph as  $G = (V, E)$ , which is assumed to be connected. Note that in the literature, FJQN/B can be cyclic, as in [13, 14]. In this paper, however, we restrict ourselves to the acyclic case.

Define the set of immediate server predecessors of server  $i \in V$ ,  $p(i)$ , to be the set of servers that have a direct link to  $i$ ,

$$p(i) = \{k \in V \mid (k, i) \in E\}$$

and the set of immediate server successors (or, the downstream servers) of server  $i \in V$ ,  $s(i)$ , to be the set of servers to which  $i$  has direct links,

$$s(i) = \{k \in V \mid (i, k) \in E\}.$$

We extend both definitions to a subgraph  $G' = (V', E')$  of  $G$ , namely,

$$p(G') = \{k \in V \setminus V' \mid (k, i) \in E\}$$

$$s(G') = \{k \in V \setminus V' \mid (i, k) \in E\},$$

where a *subgraph* in  $G$ ,  $G' = (V', E')$  is composed of a subset of nodes  $V' \subseteq V$  and edges  $E' = \{(i, j) : i, j \in V' \wedge (i, j) \in E\}$ .

The blocking FJQN/B behaves as follows. Server  $i$  initiates a *service period* whenever there resides at least one job in each of the upstream buffers and there is space for at least one job in each of the downstream buffers. Server  $i$  is said to be *starved* if at least one of the immediate upstream buffers is empty and *blocked* if at least one of the immediate downstream buffers is full. Note that the server can simultaneously be starved and blocked. Jobs remain in its upstream buffers,  $p(i) \times \{i\}$ , throughout the service period, i.e., there is no space associated with the servers for storing jobs. At the completion of the service period, a job is removed from each of its upstream buffers and a job is immediately placed in each of the downstream buffers  $\{i\} \times s(i)$ . Observe that the blocking mechanism described above corresponds to what is referred to as *blocking before service* in the literature [13].

There may be some servers for which there are no incoming edges. Each such server is referred to as a *source*, and is assumed to have an infinite number of jobs available to it so that it is never starved. There may be other servers for which there are no outgoing edges. Each such server is referred to as a *sink*, and is assumed to never be blocked. Each job that completes at a sink leaves the system immediately.

An example of a FJQN/B is given in Figure 4 (servers are represented by circles and buffers by rectangles). For convenience, we omit the drawing of the buffers in all graphs in the rest of the paper, simply assuming there is a finite capacity buffer  $B_{i,j}$  associated with each edge  $(i, j) \in E$ .

Let  $\mathbf{m}(t) = (m_{i,j}(t) : (i, j) \in E)$  be the *marking* of the system at time  $t \geq 0$  where  $m_{i,j}(t)$  denotes the number of jobs in buffer  $(i, j) \in E$  at time  $t$ . The initial marking at time  $t = 0$  is assumed to be  $\mathbf{m}(0) = (0, \dots, 0)$ .

The durations of the service periods at server  $i$  are given by a sequence of non-negative service times,  $\{\sigma_{i,n}\}_{n \geq 1}$ ,  $i \in V$ . We note that service times may take value zero. The introduction of servers whose service periods are of zero length is useful for modeling synchronization mechanisms. We will assume that each such sequence is i.i.d. and that they are mutually independent among servers. Last, we assume the existence of an unbounded rv,  $\sigma$ , that has finite mean and that stochastically bounds  $\sigma_{i,n}$  from above.

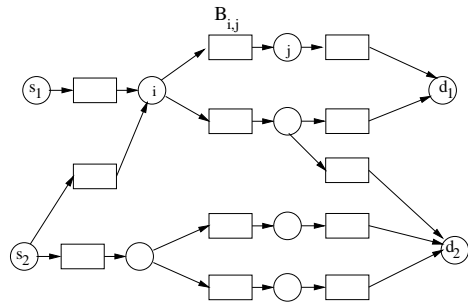


Figure 4: Example of a FJQN/B.

The performance measure of interest to us is the system throughput which we introduce as follows.

Let  $R_{i,n}(\mathcal{N})$  denote the time of the  $n$ -th service completion at server  $i$  in  $\mathcal{N}$ . The evolution equations are:

$$R_{i,1}(\mathcal{N}) = \sigma_{i,1} + \max \left\{ \max_{j \in p(i)} R_{j,1}(\mathcal{N}), \max_{k \in s(i)} R_{k,1-B_{i,k}}(\mathcal{N}) \right\}, (1)$$

$$R_{i,n}(\mathcal{N}) = \sigma_{i,n} + \max \left\{ \max_{j \in p(i)} R_{j,n}(\mathcal{N}), R_{i,n-1}(\mathcal{N}), \max_{k \in s(i)} R_{k,n-B_{i,k}}(\mathcal{N}) \right\}, \forall i \in V, n \geq 2, (2)$$

where, by convention,  $R_{i,n} = 0$ ,  $n \leq 0$ .

Let

$$R_n = \max_{i \in V} R_{i,n}.$$

Denote by  $\theta_i(\mathcal{N})$  the (asymptotic) throughput of server  $i \in V$  and  $\theta(\mathcal{N})$  the (asymptotic) throughput of  $\mathcal{N}$ . The throughput can be expressed as

$$\theta_i(\mathcal{N}) = E \left[ \left( \lim_{n \rightarrow \infty} \frac{R_{i,n}(\mathcal{N})}{n} \right)^{-1} \right], \quad i \in V, (3)$$

$$\theta(\mathcal{N}) = E \left[ \left( \lim_{n \rightarrow \infty} \frac{R_n(\mathcal{N})}{n} \right)^{-1} \right], (4)$$

provided the limits exist. It has been shown in [13] that the throughput defined above exists and is unique when the FJQN/B is deadlock free, and the service times of the servers are jointly stationary and ergodic, and have finite means.

We are interested in the scalability properties of FJQN/Bs. Specifically, we focus on determining conditions under which the throughput of a FJQN/B remains bounded away from zero as we scale up its size. In order to do so, we find it useful to introduce some graph theoretic concepts.

Consider a directed graph  $G = (V, E)$ . We define a *path* from node  $i$  to node  $j$  to be a sequence of contiguous edges, denoted by  $i = i_0 \rightarrow i_1 \rightarrow \dots \rightarrow i_k = j$ , such that  $(i_0, i_1), (i_1, i_2), \dots, (i_{k-1}, i_k)$  are in  $E$ . A *chain* between  $i$  and  $j$  is a sequence of undirected contiguous edges, denoted by  $i = i_0 \leftrightarrow i_1 \leftrightarrow \dots \leftrightarrow i_k = j$ , such that either  $(i_l, i_{l+1}) \in E$  or  $(i_{l+1}, i_l) \in E$ ,  $l = 0, \dots, k-1$ . A path (resp. chain) is *loop-free* if each node occurs in the path (resp. chain) at most once. A subgraph  $G'$  is said to be *complete* if there exists no (loop free) chains between two nodes,  $i, j \in V'$  that contains nodes not in  $G'$ .

*Definition 1:* A directed acyclic graph is said to be *strongly connected (SC)* if, for every pair of nodes in  $V$ , there exist at least two chains in the graph that do not share any nodes except for the starting and terminating nodes.

The graph in Figure 5, for example, is not strongly connected, however, the subgraph composed by nodes 1, 2, 3, 4, 5 is complete and strongly connected.

A subgraph  $G'$  is said to be a *maximal strongly connected subgraph* of  $G$  if it is strongly connected and is not contained in a larger strongly connected subgraph of  $G$ . Note that if  $G$  contains two or more maximal strongly connected subgraphs, these subgraphs share no edges in common. They may share a node. Consider the graph in Figure 5 for example; it contains two maximal SC subgraphs: the subgraph composed of nodes 1, 2, 3, 4, 5, and the subgraph of nodes 5, 6, 7, 8. Note that they have one node 5 in common.

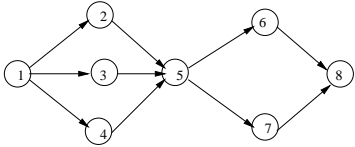


Figure 5: Maximal SC subgraphs.

### 3. PRELIMINARIES

In this section, we present preliminary properties of general FJQN/Bs and existing results on FJQN/Bs of tandem and tree topologies.

#### 3.1 Stochastic Comparison Properties

Let  $\mathcal{L}$  be a set of functions such that  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  for  $f \in \mathcal{L}$ . We say that random variable  $X$  is smaller than random variable  $Y$  in the sense of partial ordering  $\leq_{\mathcal{L}}$ , denoted  $X \leq_{\mathcal{L}} Y$ , iff for any function  $f \in \mathcal{L}$ ,  $E[f(X)] \leq E[f(Y)]$ . The partial ordering used in this paper are the basic *stochastic ordering* (when  $\mathcal{L}$  is the set of nondecreasing functions), the *convex ordering* (when  $\mathcal{L}$  is the set of convex functions), and the *increasing convex ordering*, (when  $\mathcal{L}$  the set of nondecreasing convex functions), respectively denoted by  $\leq_{st}$ ,  $\leq_{cx}$ , and  $\leq_{icx}$  (see [32] for various properties and applications of stochastic ordering).

The following stochastic comparison properties of FJQN/B's have been established in [6, 14].

LEMMA 1. Consider a FJQN/B  $\mathcal{N} = (V, E, B)$  with i.i.d. service times  $\{\bar{\sigma}(n)\}_n := \{\sigma_{i,n}; i \in V\}_{n=0}^{\infty}$ . We have the following stochastic comparison properties:

- Reducing buffer sizes only reduces throughput, i.e.  $\theta(\mathcal{N}; B+k) \geq \theta(\mathcal{N}; B), \forall k \geq 0$ .
- Increasing service times only decreases throughput, i.e. if  $\{\bar{\sigma}(n)\}_n \leq_{icx} \{\bar{\sigma}'(n)\}_n$ , then  $\{D_n\}_n \leq_{icx} \{D'_n\}_n$ , and  $\theta(\mathcal{N}) \geq \theta'(\mathcal{N})$ .
- The more constrained (in nodes and edges) is the network, the less is the throughput, i.e. for  $\mathcal{N} = (V, E, B)$  and  $\bar{\mathcal{N}} = (\bar{V}, \bar{E}, B)$  with  $V \subset \bar{V}$ ,  $E \subset \bar{E}$ , we have  $\{D_n\}_n \leq_{st} \{\bar{D}_n\}_n$ , and  $\theta(\mathcal{N}) \geq \theta(\bar{\mathcal{N}})$ .

Based on the Lemma 1, in order to show that a general FJQN/B system is throughput scalable, it suffices to show that the corresponding homogeneous FJQN/B system (with all buffer sizes equal to  $\min_i B_i$  and all service times i.i.d. having the same distribution as the upper bound r.v.  $\sigma$ ) is throughput scalable.

The following two lemmas will be useful for later sections. Detailed proofs are deferred to the appendix.

LEMMA 2. Suppose  $X_i$ 's  $i = 0, 1, \dots$ , are i.i.d. random variables. Then

$$\frac{X_1 + X_2 + \dots + X_n}{n} \leq_{cx} X_0.$$

LEMMA 3. Consider a FJQN/B  $\mathcal{N}$  with two different sets of i.i.d. service time sequences,  $\{\bar{\sigma}(n)\}_n$  and  $\{\bar{\sigma}'(n)\}_n$ . If there is a constant  $\gamma > 0$ , such that  $\bar{\sigma}(n) \stackrel{d}{=} \gamma \cdot \bar{\sigma}'(n)$ , for all  $n$ , then  $\theta(\mathcal{N}; \sigma) = \frac{1}{\gamma} \theta(\mathcal{N}; \sigma')$ .

The following result on the duality and reversibility properties of FJQN/B's has been obtained in [13].

LEMMA 4. Consider a FJQN/B with i.i.d. service times.

- Duality:** Define the dual FJQN/B by reversing the direction of the flow and changing the corresponding initial marking to the number of holes initially present. Then the dual and the original FJQN/B have the same throughput.
- Reversibility:** Define the reverse FJQN/B to be the network obtained by reversing the flows of jobs while keeping the same initial marking, the reverse and the original FJQN/B have the same throughput.

We say that a real random variable  $X$  is *light tailed* if there exists  $a > 0$  such that  $E[e^{aX}] < \infty$ ; otherwise,  $X$  is said to be *heavy tailed*.

Throughout the paper, we denote  $a_n = o(b_n)$  if  $a_n/b_n \rightarrow 0$  as  $n \rightarrow \infty$ ; and  $a_n \sim b_n$  if  $a_n/b_n \rightarrow r$  as  $n \rightarrow \infty$  for some constant  $r > 0$ . We will also need the following results from extreme value theory [18].

LEMMA 5. Denote  $M_n := \max_{1 \leq i \leq n} X_i$ , where  $X_i$ 's are i.i.d. random variables with distribution function  $G$ . Let  $\bar{G} = 1 - G$ , and  $b_n = (\bar{G})^{-1}(\frac{1}{n})$ . Then

$$E[M_n] \sim b_n. \quad (5)$$

Hence for light-tailed  $X$ 's with exponential-tail  $\bar{G} = e^{-x}$ ,  $b_n = \ln n$ ; and for heavy-tailed  $X$ 's with Pareto distributions  $\bar{G} = x^{-\alpha}$ ,  $b_n = n^{\frac{1}{\alpha}}$ .

#### 3.2 Tandem and Tree Networks with Blocking

Two special cases of FJQN/B's that have received much attention are open tandem queueing networks and tree networks. Such networks are prevalent in many systems and applications. For example, the reliable end-to-end transfer in a mobile ad-hoc network can be represented as a tandem queue network with blocking. And a tree network with blocking has been used in [2] to represent the reliable multicast in a one-to-many TCP overlay network.

The following result has been established in [26] for tandem networks with blocking.

LEMMA 6. Consider an arbitrary tandem network  $\mathcal{T}$  (with blocking), where all buffers are of finite capacity  $B$ , and all service times are i.i.d. and satisfy condition

$$\int_0^{\infty} P(\sigma > x)^{1/2} dx < \infty. \quad (6)$$

The throughput  $\theta(\mathcal{T})$  is bounded below by a positive constant, independent of the network size. i.e. there exists  $\eta_0 > 0$ , such that

$$\theta(\mathcal{T}) \geq \eta_0, \quad \text{for all } \mathcal{T}.$$

Note that condition (6) is stronger than  $E[\sigma^2] < \infty$ , but slightly weaker than the requirement  $E[\sigma^{2+\epsilon}] < \infty$  for some  $\epsilon > 0$ .

Using theoretical investigations and large-network simulations, [8] reports that under some mild conditions, the throughput of a tree network with blocking is bounded below by a positive constant, independent of the network size. The result is restated as follows.

**LEMMA 7.** *Consider an arbitrary tree network  $\mathcal{TR}$  with blocking. The fan-out degree of the tree is bounded from above by a constant  $D$ , the buffer sizes are bounded below by a constant  $B$ , and the service times are independent and upper bounded (in the stochastic ordering sense) by a random variable  $\sigma$  that is light tailed. The throughput of such a network is bounded below by a positive constant, independent of the network size; i.e. there exists  $\eta_1 > 0$ , such that*

$$\theta(\mathcal{TR}) \geq \eta_1, \quad \text{for all } \mathcal{TR}.$$

In the following sections we address the throughput scalability of general FJQN/B's as the network size grows. We will simply focus on the homogeneous case where all buffers are of the same size  $B$ , and all service times are i.i.d. with the same distribution as  $\sigma$ . Similar results can be obtained for the non-homogeneous case using the monotonicity properties given by Lemma 1. We answer the scalability question by providing some useful upper and lower bounds on system throughput. These bounds then shed light on determining conditions under which the system throughput does or does not scale as the networks grow in size.

## 4. THROUGHPUT BOUNDS

In this section we obtain useful upper and lower bounds on system throughput, that can help address the scalability question. We consider a general FJQN/B  $\mathcal{N}$  that has  $N$  nodes. All service times  $\sigma$  are i.i.d. with finite mean. Let  $D$  (resp.  $L$ ) denote the length (in number of edges) of the shortest (resp. longest) path in  $\mathcal{N}$ . Denote by  $K$  the largest out-degree or in-degree of any node in the network.

A simple upper bound is obtained by comparing the system of interest with one where all service times are replaced by their averages. The result follows directly from the convex ordering that  $E\sigma \leq_{cx} \sigma$ , and Lemma 1.b).

**LEMMA 8 (DETERMINISTIC BOUND).** *Let  $\bar{\theta}(\mathcal{N})$  be the throughput of the network which has the same topology as  $\mathcal{N}$  but replacing each random service time by its mean value. Then*

$$\theta(\mathcal{N}) \leq \bar{\theta}(\mathcal{N}).$$

We also claim the following upper bound which relates to the maximum node degree in  $\mathcal{N}$ .

**THEOREM 9 (MAX DEGREE BOUND).** *For a network  $\mathcal{N}$  with maximum (out or in)-degree  $K$ ,*

$$\theta(\mathcal{N}) \leq \frac{B}{E \max_{1 \leq i \leq K} \sigma_i}. \quad (7)$$

**Proof.** Suppose  $K$  is the largest out-degree of node  $v_0 \in V$ . Denote by  $v_i$  the  $i$ -th down-stream neighbor of  $v_0$ , and  $B_{0,k}$  the buffer between nodes  $v_0$  and  $v_k$ ,  $k = 1, \dots, K$ . We then have a 'fork subsystem' composed of nodes  $v_0, v_1, \dots$ , and  $v_K$ . It suffices to study the throughput of the 'fork subsystem'.

Denote by  $\bar{Q}$  and  $\bar{R}$  the average number of jobs and average sojourn time of a job in the fork subsystem in steady state. By Little's law we have  $\theta(\text{fork}) = \bar{Q}/\bar{R}$ . We will bound  $\theta(\text{fork})$  by bounding  $\bar{Q}$  and  $\bar{R}$ . Consider the oldest job in all buffers  $B_{0,1}, \dots, B_{0,K}$ . Suppose it is job  $n$  in buffer  $B_{0,i}$ . There can be no more than  $B - 1$  jobs younger than job  $n$  in buffer  $B_{0,i}$ . Due to the forking, no job younger than these  $B - 1$  jobs can be present in any of the buffers. Therefore,  $\bar{Q} \leq B$ . On the other hand, the sojourn time of each job in the fork system is at least  $\max_{1 \leq i \leq K} \sigma_i$ . and  $\bar{R} \geq E[\max_{1 \leq i \leq K} \sigma_i]$ . Thus, the throughput of the fork system satisfies:

$$\theta(\text{fork}) = \frac{\bar{Q}}{\bar{R}} \leq \frac{B}{E[\max_{1 \leq i \leq K} \sigma_i]}.$$

The result then follows since the overall throughput can be no higher than  $\theta(\text{fork})$ .

For the case when  $K$  is the maximal in-degree of the network, one can use reversibility in Lemma 4 to convert the in-degree join with  $K$  branches into an equivalent out-degree fork with  $K$  branches, and then apply the above argument. ■

We next focus on a subgraph  $\mathcal{N}_{a,b} \subset \mathcal{N}$ , which is composed of nodes and edges on all paths between a given pair of nodes  $a, b \in V$ . We will refer to  $\mathcal{N}_{a,b}$  as a *Single-Input-Single-Output (SISO)* subgraph of  $\mathcal{N}$  since  $\mathcal{N}_{a,b}$  has a single source  $a$  and single sink  $b$ . Denote by  $N_{a,b}$  the total number of nodes,  $K_{a,b}$  the maximum node (out or in) degree, and  $\mathcal{P}_{a,b}$  the set of all paths from  $a$  to  $b$  in  $\mathcal{N}_{a,b}$ . Let  $D_{a,b}$  (resp.  $L_{a,b}$ ) denote the length of the shortest (resp. longest) path in  $\mathcal{N}_{a,b}$ . For a given path  $p \in \mathcal{P}_{a,b}$ :  $a = p_0 \rightarrow p_1 \rightarrow \dots \rightarrow p_l = b$ , denote by  $S_n^p$  the total service requirement of job  $n$  along path  $p$ , i.e.

$$S_n^p = \sum_{j=0}^l \sigma_{p_j, n}.$$

We establish the following upper and lower bounds for the throughput of  $\mathcal{N}_{a,b}$ .

**THEOREM 10.** *The throughput of any SISO subgraph  $\mathcal{N}_{a,b}$  satisfies:*

$$\frac{1}{E[\max_{p \in \mathcal{P}_{a,b}} S_1^p]} \leq \theta(\mathcal{N}_{a,b}) \leq \frac{D_{a,b} \cdot B}{E[\max_{p \in \mathcal{P}_{a,b}} S_1^p]}, \quad (8)$$

**Proof.** Observe that each job in  $\mathcal{N}_{a,b}$  needs to traverse all paths  $p \in \mathcal{P}_{a,b}$  from  $a$  to  $b$ , the minimum time a job stays in  $\mathcal{N}_{a,b}$  is  $\max_{p \in \mathcal{P}_{a,b}} S_1^p$ . A lower bound on  $\theta(\mathcal{N}_{a,b})$  can therefore be derived by allowing at most one job in the system each time, which yields the left hand side of (8).

We next prove the right hand side of (8). Since the shortest path in  $\mathcal{N}_{a,b}$  is of length  $D_{a,b}$ , there are at most  $W = D_{a,b} \cdot B$  jobs in the network at any given time. Now consider jobs  $W, 2W, 3W, \dots$ , and denote  $R_{kW}$  the service completion time of job  $kW$ . Note that job  $kW$  cannot start service at source  $a$  until job  $(k-1)W$  has left the sink  $b$ . Since job  $kW$  then needs to traverse all paths from  $a$  to  $b$ ,

we have:

$$R_{kW}(\mathcal{N}) \geq R_{(k-1)W}(\mathcal{N}) + \max_{p \in \mathcal{P}_{a,b}} S_{kW}^p. \quad (9)$$

Hence, by summing up (9) for  $k = 1, 2, \dots, m$ , dividing the sum by  $mW$ , and letting  $m$  approach infinite, we have

$$\lim_{m \rightarrow \infty} \frac{R_{mW}(\mathcal{N}_{a,b})}{mW} \geq \frac{1}{W} \lim_{m \rightarrow \infty} \frac{\sum_{k=1}^m \max_{p \in \mathcal{P}_{a,b}} S_{kW}^p}{m}.$$

Since  $\{\max_{p \in \mathcal{P}_{a,b}} S_{mW}^p\}_{m=1,2,\dots}$  are i.i.d. random variables, application of the law of large numbers yields the right hand side of (8). ■

The result of Theorem 10 can be further generalized to Single-Input-Multiple-Output(SIMO) or Multiple-Input-Single-Output(MISO) subgraphs of  $\mathcal{N}$ . Denote  $V_o$  and  $V_d$  respectively the set of sources and sinks in  $\mathcal{N}$ . A SIMO subgraph of  $\mathcal{N}$  consists of nodes and edges on all paths between a single source in  $V_o$  and all sinks in  $V_d$  that are reachable from that source. Similarly, a MISO subgraph of  $\mathcal{N}$  consists of nodes and edges on all paths between a single sink in  $V_d$  and all sources in  $V_o$  that can reach that sink. We claim the following result without proof (since the proof is essentially the same as that of Theorem 10).

**THEOREM 11.** *Consider an arbitrary SIMO, MISO or SISO subgraph of  $\mathcal{N}$ , denoted by  $\mathcal{N}_s$ . Its throughput must satisfy*

$$\frac{1}{E[\max_{p \in \mathcal{P}_s} S_1^p]} \leq \theta(\mathcal{N}_s) \leq \frac{D_s \cdot B}{E[\max_{p \in \mathcal{P}_s} S_1^p]}, \quad (10)$$

where  $\mathcal{P}_s$  and  $D_s$  denote respectively the set of all paths and the length of the shortest path in  $\mathcal{N}_s$ .

For convenience, we denote  $\mathcal{F}_s$  the family of all SISO, SIMO, and MISO subgraphs in  $\mathcal{N}$ . Note that an SISO subgraph can be between two arbitrary nodes in  $V$ , not necessarily just from a source to a sink. Consider the graph in Figure 5 for example; the subgraph composed of nodes 1, 2, 3, 4, 5, the subgraph of nodes 5, 6, 7, 8, and the whole graph itself, are all examples of SISO subgraphs.

Combining Theorems 9 and 11, and the extreme value theory results from Lemma 5 leads to the following corollary.

**COROLLARY 12.** *Consider an arbitrary  $\mathcal{N}_s \in \mathcal{F}_s$ , with size  $N_s$ , maximum out- or in-degree  $K_s$ , shortest and longest path lengths  $D_s$  and  $L_s$ . There exist constants  $C_1$  and  $C_2$  such that,*

$$\frac{C_1}{N_s} \leq \theta(\mathcal{N}_s) \leq C_2 \cdot \min \left( \frac{1}{\log K_s}, \frac{D_s}{L_s}, \frac{D_s}{\log N_s} \right), \quad (11)$$

when the service times are light-tailed; and

$$\frac{C_1}{N_s} \leq \theta(\mathcal{N}_s) \leq C_2 \cdot \min \left( K_s^{-1/\alpha}, \frac{D_s}{L_s}, D_s N_s^{-1/\alpha} \right), \quad (12)$$

when the service times are heavy-tailed, with Pareto distribution  $\overline{G} = x^{-\alpha}$ ,  $\alpha > 1$ .

**Proof.** The result follows from (7) and (10), and by applying extreme value theory with the following two facts:

- 1)  $E[\max_{p \in \mathcal{P}_s} S_1^p] \geq E[S_1^{\text{longestpath}}] = L_s \cdot E[\sigma]$ ; and
- 2)  $E[\sum_{i=1}^{N_s} \sigma_{i,1}] \geq E[\max_{p \in \mathcal{P}_s} S_1^p] \geq E[\max_{i=1}^{N_s} \sigma_{i,1}]$ . ■

Since the original network  $\mathcal{N}$  has more node and edge constraints than any of its subnetworks, from Lemma 1, we have  $\theta(\mathcal{N}) \leq \theta(\mathcal{N}_s)$  for all  $\mathcal{N}_s \in \mathcal{F}_s$ . Based on the upper bound results from (11) and (12), we observe that the throughput

of  $\mathcal{N}$  can decrease to zero when the number of branches goes to infinity, or when the network is not well-balanced in the sense that the ratio of the shortest and the longest paths between any pair of nodes goes to zero. We next define more rigorously the requirement on such path balance in order to make a network throughput scalable.

**Definition 2:** A FJQN/B  $\mathcal{N} = (V, E, B)$  is said to be  $c$ -balanced if the ratio of the number of buffers (edges) for any pair of paths between all pairs of nodes is bounded from above by a constant  $c$  (with  $0 < c < \infty$ ), in other words, for every pair of paths  $p_1$  and  $p_2$  between any pair of nodes  $v_1, v_2 \in V$ ,

$$1/c \leq N(p_1)/N(p_2) \leq c.$$

where  $N(p)$  denotes the number of servers on path  $p$ . We call the smallest  $c$  that satisfies the above condition the *balance-ratio* of  $\mathcal{N}$ .

We then have the following upper bounds on the throughput of a general FJQN/B  $\mathcal{N}$ .

**COROLLARY 13.** *Consider a FJQN/B network  $\mathcal{N}$ , with  $N$  nodes, max out- (or in-) degree  $K$ , shortest path length  $D$ , balance-ratio  $c$ , and i.i.d. service times with finite mean. There exists a constant  $C$  such that*

$$\theta(\mathcal{N}) \leq C \cdot \min \left( \frac{1}{\log K}, \frac{1}{c}, \min_{\mathcal{N}_s \in \mathcal{F}_s} \frac{D_s}{\log N_s} \right),$$

when the service times are light-tailed; and

$$\theta(\mathcal{N}) \leq C \cdot \min \left( K^{-1/\alpha}, \frac{1}{c}, \min_{\mathcal{N}_s \in \mathcal{F}_s} D_s N_s^{-1/\alpha} \right),$$

when the service time are heavy-tailed with Pareto distribution  $\overline{G} = x^{-\alpha}$ ,  $\alpha > 1$ .

We are now ready to state necessary conditions for a FJQN/B network to have a scalable throughput.

**COROLLARY 14 (NECESSARY CONDITIONS).** *The following conditions are necessary for a FJQN/B network to have its throughput bounded away from zero:*

- a). *all nodes must have finite (in and out) degrees, i.e.  $|p(i)| \leq K$ , and  $|s(i)| \leq K$  for all  $i \in V$ ;*
- b). *it must be  $c$ -balanced for some constant  $c$ ;*
- c). *there is a positive constant  $\eta > 0$  such that for all  $\mathcal{N}_s \in \mathcal{F}_s$ ,  $\frac{D_s}{\log N_s} \geq \eta > 0$  if the service times are light-tailed; and  $D_s N_s^{-\frac{1}{\alpha}} \geq \eta > 0$  if the service times are heavy tailed with Pareto distribution  $\overline{G} = x^{-\alpha}$ .*

To obtain sufficient conditions for scalability, we need better lower bounds. Such lower bounds are derived in the next two sections for two special families of networks. Our main idea of achieving such lower bounds is to bound the original network from below by either a tandem network or a tree network, utilizing the fact that both have been proved to be throughput scalable.

## 5. SERIES-PARALLEL NETWORKS

In this section, we consider the special family of series-parallel networks. We provide conditions that guarantee the scalability of such networks, and also show the exact order of degradation when the network throughput decays to zero.

Throughout the rest of the paper, we assume that:

**Assumption 1:** *The service times are i.i.d. at each server, independent of the services at other servers. We denote by  $\sigma$  the generic service time and assume that condition (6) is satisfied.*

We start by considering a two-branch series-parallel (SP) network, denoted by  $\mathcal{N}_N$ , as shown in Figure 6.  $N$  represents the total number of nodes in the network. There is one source node  $s$  and one sink node  $d$ . Denote  $D_N$  and  $L_N$  respectively the lengths of the shorter and the longer paths, with  $D_N \leq L_N$ . All buffers are of size  $B$  and all service times are i.i.d. with finite mean.

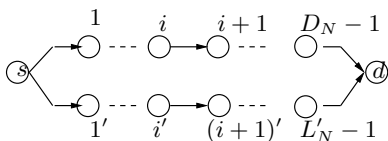
**THEOREM 15.** *The throughput of a two-branch SP FJQN/B network  $\mathcal{N}_N$  is bounded below by a positive constant  $\eta > 0$ , i.e.*

$$\theta(\mathcal{N}_N) \geq \eta > 0,$$

*independent of its size  $N$ , if there exists  $c < \infty$ , such that*

$$\lim_{N \rightarrow \infty} \frac{L_N}{D_N} = c. \quad (13)$$

*If  $c = \infty$ , then  $\theta(\mathcal{N}_N) \rightarrow 0$  as  $N \rightarrow \infty$ .*



**Figure 6: A Series-parallel graph with two branches**

**Proof.** We need a lower bound for the throughput. We can restrict to the case  $D_N = L_N$  and  $B = 1$ . Denote the nodes (except the source and sink) in the top branch by  $i$  and nodes in the lower branch by  $i'$ ,  $i = 1, \dots, D_N - 1$ . First, we introduce edges between nodes  $i'$  and  $i + 1$ , and between nodes  $i$  and  $(i + 1)'$  for  $i = 1, \dots, D_N - 1$ . It follows from Lemma 1 that the throughput is lower for this more constrained system than the original system. In this new system, a job can only proceed to server  $i + 1$  (or server  $(i + 1)'$ ) when it has completed services on both nodes  $i$  and  $i'$ . It is easy to see that this new system is sample path equivalent to a tandem system in which nodes  $i$  and  $i'$  are replaced by a single (super)node  $i$  with incoming edge  $(i - 1, i)$  and outgoing edge  $(i, i + 1)$  with buffer sizes one, and service time  $\sigma_{i,n} = \max\{\sigma_{i,n}, \sigma_{i',n}\}$ . Since the tandem network throughput remains bounded away from zero for all sizes, then based on Lemma 6, the original system throughput must also scale.

If instead,  $L_N = cD_N$ ,  $c > 1$ , we can similarly combine every  $c$  nodes in the longer branch with one node in the shorter branch and replace the group by a supernode, with service time  $\sigma^S \stackrel{d}{=} \max(\sum_{i=1}^c \sigma_i, \sigma'_1)$ . We then obtain a tandem network, which has a lower throughput than the original system. As long as  $c$  is finite, the supernode service time stays finite, and the resulting tandem network throughput stays bounded away from zero based on Lemma 6. ■

We can further generalize the results of Theorem 15 to arbitrary  $K$ -branch (with fixed  $K < \infty$ ) SP FJQN/B networks, using a similar argument to bound the network from below by a tandem queueing network. Combine with the upper bound developed in the previous section, we then have the necessary and sufficient condition.

**COROLLARY 16** (NECESSARY AND SUFFICIENT CONDITION). *Condition (13) is necessary and sufficient for an arbitrary  $K$ -branch SP FJQN/B network to stay throughput scalable independent of its size.*

If the throughput does not scale, i.e. when  $D_N = o(L_N)$ , we can further show the throughput degradation in exact order, which is given by the next theorem.

**THEOREM 17.** *The throughput of a  $K$ -branch SP FJQN/B network satisfies:*

$$\theta(\mathcal{N}_N) \sim \frac{D_N}{L_N} \cdot \theta(\mathcal{T}_{D_N}), \quad \text{for } D_N = o(L_N),$$

*where  $\mathcal{T}_N$  denotes a  $N$ -stage tandem queueing network with blocking with i.i.d. service times  $\sigma$ .*

**Proof.** Based on (10), an upper bound on  $\theta(\mathcal{N}_N)$  is:

$$\theta(\mathcal{N}_N) \leq C \cdot \frac{D_N}{L_N} \sim \frac{D_N}{L_N}. \quad (14)$$

Denote  $c_N = \lceil (L_N - 1)/(D_N - 1) \rceil$ . As mentioned earlier, a lower bound can be obtained by grouping every  $c_N$ -nodes in the longer branch with 1 node in the shorter branch and replacing the cluster by a supernode with service time  $\sigma^S \stackrel{d}{=} \max(\sum_{i=1}^{c_N} \sigma_i, \sigma'_1)$ . We then have a  $D_N$ -stage tandem network, denoted by  $\mathcal{T}_{D_N}^S$ .

We introduce further a normalized tandem system  $\mathcal{T}_{D_N}^o$ , which is derived from  $\mathcal{T}_{D_N}^S$  but with normalized service times  $\sigma^o = \frac{\sigma^S}{c_N + 1}$ . Apply Lemma 3, we then have  $\theta(\mathcal{T}_{D_N}^o) = (c_N + 1)\theta(\mathcal{T}_{D_N}^S)$ .

Based on Lemma 2, we know that

$$\sigma^o = \frac{\sigma^S}{c_N + 1} \leq \frac{\sum_{i=1}^{c_N} \sigma_i + \sigma'_1}{c_N + 1} \leq c_x \sigma.$$

It follows from Lemma 1.b) that  $\theta(\mathcal{T}_{D_N}^o) \geq \theta(\mathcal{T}_{D_N})$ , where  $\mathcal{T}_{D_N}^o$  and  $\mathcal{T}_{D_N}$  denote the throughput of a  $D_N$ -stage tandem queueing network respectively with i.i.d. service times  $\sigma^o$  and  $\sigma$ .

Therefore,

$$\theta(\mathcal{N}_N) \geq \theta(\mathcal{T}_{D_N}^S) = \frac{1}{c_N + 1} \theta(\mathcal{T}_{D_N}^o) \geq \frac{1}{c_N + 1} \theta(\mathcal{T}_{D_N}). \quad (15)$$

The result then follows from (14), (15) and Lemma 6. ■

We can further generalize the throughput degradation results of Theorem 17 for two branch SP networks to arbitrary  $K_N$ -branch SP FJQN/B networks. The proof is in similar spirit as that of Theorem 17 and thus omitted.

**THEOREM 18.** *Consider a sequence of Series-Parallel networks  $\mathcal{N}_N$ , where  $\mathcal{N}_N$  is of size  $N$ , with  $K_N$ -branches, shortest path length  $D_N$ , and longest path length  $L_N$ .*

- If  $L_N \leq M < \infty$ , then

$$\theta(\mathcal{N}_N) \sim \frac{1}{\log K_N}, \quad \text{for large } K_N,$$

*if the service times are i.i.d. light-tailed; and*

$$\theta(\mathcal{N}_N) \sim K_N^{-1/\alpha}, \quad \text{for large } K_N,$$

*if the service time are heavy-tailed with Pareto distribution  $\overline{G} = x^{-\alpha}$ ,  $\alpha > 1$ .*

- If  $K_N \leq M < \infty$ , then

$$\theta(\mathcal{N}_N) \sim \frac{D_N}{L_N}, \quad \text{for } D_N = o(L_N).$$

## 5.1 Scalability of Closed Queueing Networks

Our results on SP FJQN/B networks can also be applied to understand the asymptotic behavior of some large (non-fork-join type) queueing networks with finite buffers. Note that analyzing such large queueing networks with finite buffers is a difficult problem as there are no explicit solutions such as the product-form solutions of Jackson networks.

Consider, for example, a two-branch series-parallel network with initial marking 0 (no job present in any buffers initially). All buffers are of size  $B$ . A dual of this network can be obtained by reversing the direction of flow in the shorter branch, and replacing all holes with jobs in that branch. We then have a closed queueing network with finite buffers, where there are  $D_N \cdot B$  jobs circulating in the system.

Based on the duality property of Lemma 4, the dual network has the same throughput as the original network. Based on the result of Theorem 17, we then obtain the following result on the asymptotic system throughput of a closed queueing network with finite buffers.

**COROLLARY 19.** *Consider a closed-queueing network CQN with  $J$  single server queues. All buffers are of size  $B$ . The service times are i.i.d. with finite mean. There are  $M$  jobs circulating in the system. The throughput must satisfy,*

$$\theta(CQN) \sim \begin{cases} \frac{M}{J}/(B - \frac{M}{J}), & M/J \leq B/2, \\ \frac{J-M}{J}/(B - \frac{J-M}{J}), & M/J > B/2, \end{cases} \quad \text{for large } J.$$

The above result is consistent with the result for an infinite-buffered closed QN with exponential service times, in which case the throughput is  $M/(J + M - 1)$ .

## 6. W-WIDE GRAPHS

Based on the results from the preceding section, we observe that throughput can decrease to zero in a series parallel graph as the number of branches goes to infinity and that there is a need to maintain a balance in the number of servers in each branch. Learning from these observations, we extend Theorem 15 to a larger class of FJQN/B's, namely those whose underlying graphs are  $w$ -wide and  $c$ -balanced. To show this general class of FJQN/B networks is scalable in throughput, we provide reduction algorithms that can bound the network from below by either a tandem queueing network or a tree network, which we know are throughput scalable. We assume that all buffers are bounded from below by a constant  $B$ , and all service times are i.i.d. with finite mean.

*Definition 3:* A directed acyclic graph(DAG)  $G = (V, E)$  is said to have a  $w$ -wide SC component  $G_c = (V_c, E_c)$  if there exists a set of at most  $w$  nodes,  $V' \subseteq V_c$ , such that there is no path between any pair of nodes in  $V'$ . We say that  $G$  is a  $w$ -wide graph if it contains at least one  $w$ -wide SC component, and there exists no  $(w + 1)$ -wide SC component in  $G$ .

A series-parallel graph with  $w$  branches is an example of a  $w$ -wide graph.

The next lemma establishes that for a graph  $G$ , if we replace a complete subgraph  $G_s$  by a supernode with service time equal to the total service requirement on all nodes in  $G_s$ , the throughput of the resulting new graph  $G'$  will be worse than that of the original graph  $G$ .

**LEMMA 20.** *Consider a FJQN/B  $\mathcal{N}$  with underlying graph  $G = (V, E)$ , service times  $\{\sigma_{v,n}, v \in V\}_n$  and identical buffer size  $B$ . Let  $G_s = (V_s, E_s)$  be a complete subgraph of  $G$ . Let  $\mathcal{N}'$  be a FJQN/B with graph  $G' = (V', E')$  such that*

$$\begin{aligned} V' &= (V \setminus V_s) \cup \{V_s\}, \\ E' &= \{(v, w) : v, w \in V \setminus V_s, (v, w) \in E\} \\ &\quad \cup \{(v, V_s) : v \in p(G_s)\} \\ &\quad \cup \{(V_s, v) : v \in s(G_s)\}, \\ \sigma'_{v,n} &= \begin{cases} \sigma_{v,n}, & v \in V \setminus V_s, \\ \sum_{v \in V_s} \sigma_{v,n}, & v = V_s \end{cases} \end{aligned}$$

and all buffers of size  $B$ . Then

$$R'_{V_s, n} \geq R_{v, n}, n = 1, 2, \dots; \forall v \in V_s,$$

and

$$\theta(\mathcal{N}) \geq \theta(\mathcal{N}').$$

**Proof.** The result follows from  $R_{v,n} \leq R'_{v,n}$ ,  $v \in V \setminus V_s$ , and  $R_{v,n} \leq R'_{V_s, n}$ ,  $v \in V_s$ , for  $n = 1, \dots, J$ , which we will show next. We do so by constructing a new FJQN/B with graph  $G^l = (V^l, E^l)$  where  $v^l = V$ . In order to define the edge set, we topologically sort the nodes in  $V_s$  according to the set of edges in  $E_s$ . There may not be a unique topological sort of these nodes. Pick one and label the nodes  $v_1, v_2, \dots, v_{|V_s|}$ . We now introduce the edge set as

$$\begin{aligned} E^l &= E \cup \{(v, w) : v \in p(G_s), w \in V_s\} \\ &\quad \cup \{(v, w) : w \in s(G_s), v \in V_s\} \\ &\quad \cup \{(v_i, v_j) : i < j\}, \end{aligned}$$

buffer sizes all equal to  $B$ , and

$$\{\sigma'_{v,n}, v \in V\}_n = \{\sigma_{v,n}, v \in V\}_n.$$

From Lemma 1, it follows that the departure times from all of the nodes can only decrease. Moreover, because of the addition of edges according to the topological sort, only one customer can receive service at any one time from any server within  $V_s$  and it does so by traversing the sequence of nodes  $v_1, \dots, v_{|V_s|}$ . The time to do so for the  $n$ -th customer is  $\sum_{v \in V_s} \sigma_{v,n}$ . Thus this system behaves exactly like  $\mathcal{N}'$  and the lemma follows. ■

The following lemma will also prove useful.

**LEMMA 21.** *Let  $\mathcal{N}$  be a FJQN/B whose underlying graph consists of an in-tree feeding an out-tree. In other words, there exists a node  $v \in V$  such that all predecessors of  $v$  have out-degrees of one and arbitrary in-degrees (0, 1, or greater), and such that all successors of  $v$  have in-degrees of one and arbitrary out-degrees (0, 1, or greater). Let  $\mathcal{N}'$  be a FJQN/B that differs from  $\mathcal{N}$  in that all upstream edges to  $v$  have been reversed and all upstream buffers have been filled with customers. Then  $\theta(\mathcal{N}) = \theta(\mathcal{N}')$ .*

**Proof.** We rely on the fact that  $\mathcal{N}$  is a fork join queueing network with blocking before service (FJQN/B). We make use of the duality property of such networks based on Lemma 1, namely that the network obtained by reversing the upstream edges to  $v$  and filling all of the buffers in this part of the network with customers, the resulting network has the same sample path behavior and, thus, the same throughput as the original network. We now operate this new network,  $\mathcal{N}'$  in the following manner, we shut off arrivals to the system and run it sufficiently long so that the

customers initially present in the system depart leaving an empty system. As soon as this occurs, we then allow users to enter the system at node  $v$ . As the time required to empty the system is finite, it follows that  $\theta(\mathcal{N}) = \theta(\mathcal{N}')$ . ■

We first consider the case where  $\mathcal{N}$  is a *single input single output (SISO) SC network*.

**THEOREM 22.** *A SISO  $w$ -wide,  $c$ -balanced, SC FJQN/B  $\mathcal{N}$  has a throughput  $\theta(\mathcal{N}) \geq \eta(w, c, \sigma) > 0$ , where  $\eta(w, c, \sigma)$  is a constant that only depends on  $w, c$  and  $\sigma$ .*

**Proof.** We can assume that all of the buffers are of size one as this provides a lower bound on the throughput for any FJQN/B based on Lemma 1. We will construct a tandem network  $\mathcal{N}^t = (V^t, E^t, \sigma^t)$  that achieves a lower throughput than network  $\mathcal{N}$ .

Before we describe the construction, we introduce some notation. Denote by  $s$  and  $d$  the source and sink of  $\mathcal{N}$ . Let  $D$  denote the number of edges on the shortest path between  $s$  and  $d$ . Let  $l(v)$  denote the number of hops on the longest path between  $v \in V$  and  $d$ .

The reduction algorithm is as follows.

**Reduction Algorithm:**

*Initialize.*  $R_0 = \{s\}$ ;  $V^t = \{R_0\}$ ;  $E^t = \emptyset$ ;  $S = V \setminus \{s\}$ ;  $i = 0$ ;  
while  $S \neq \emptyset$  do  
     $i = i + 1$   
     $R_i = \{v : v \in s(R_{i-1})\}$   
     $\cup \{v \in V : (D - i + 1)c > l(v) \geq (D - i)c\}$ ;  
    add ancestors of nodes in  $R_i$  not in  $\cup_{k=0}^i R_k$   
    to  $R_i$ ;  
     $S = S \setminus R_i$ ;  
     $\sigma_{i,n}^t = \sum_{v \in R_i} \sigma_{v,n}$ ;  
     $V^t = V^t \cup \{R_i\}$ ;  
     $E^t = E^t \cup \{(R_{i-1}, R_i)\}$   
endwhile

With the above construction, we claim that  $\mathcal{N}^t$  exhibits three important properties.

- 1). If  $(v_1, v_2) \in E$ , then either  $v_1, v_2 \in R_i$ , for some  $i = 1, \dots, D$ , or  $v_1 \in R_i, v_2 \in R_{i+1}, i = 0, \dots, D$ .
- 2). Each  $R^i$  is a complete subgraph of  $G$ .
- 3).  $|R_i| \leq wc$  for  $i = 1, \dots, D$ .

To show 1), suppose this is not true, i.e., there exists a  $(v_1, v_2) \in E$  such that  $v_1 \in R_i, v_2 \in R_j$ , and  $j \neq i + 1$ . Suppose that  $i < j$ . This cannot occur because our construction of  $R_{i+1}$  includes all downstream neighbors of nodes in  $R_i$  that are not in  $R_i$ . Suppose that  $i > j$ . This cannot occur because our construction ensures that all ancestors of nodes in  $R_j$  lie either within  $R_j$  or  $R_k, k < j$ .

Property 2) follows immediately from 1). This is because if there is a path between two nodes  $v_1, v_2 \in R_j$ , all nodes on the path must also be in  $R_j$ , since all ancestors of nodes in  $R_j$  lie either within  $R_j$  or  $R_k, k < j$ .

Property 3) follows from the assumption that the original network is both  $c$ -balanced and  $w$ -wide.

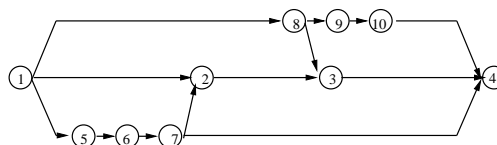
Finally, it follows from Lemma 20 that the departure times  $\{R_{i,n}^t\}$  for  $\mathcal{N}^t$  satisfy

$$R_{i,n}^t \geq R_{v,n}, \quad n = 1, \dots; \quad v \in R_i$$

From this we conclude that  $\mathcal{N}^t$  is a tandem network whose service times are stochastically bounded from above by the  $wc$ -fold convolution of  $\sigma$  henceforth referred to as  $\sigma^{(wc)}$ .

Hence Lemma 6 applies here yielding the desired results. ■

Consider, for example, the graph in Figure 7. The algorithm will result in  $R_0 = \{1\}, R_1 = \{2, 5, 6, 7, 8\}$ , and  $R_2 = \{3, 4, 9, 10\}$ .



**Figure 7: An Example**

Next, we consider a *multiple input multiple output (MIMO) FJQN/B*.

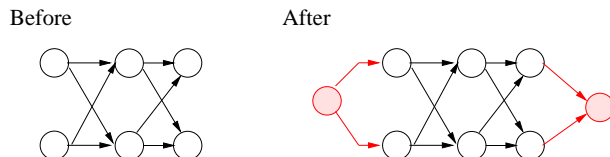
**THEOREM 23.** *A MIMO,  $w$ -wide,  $c$ -balanced, SC FJQN/B  $\mathcal{N}$  has a throughput  $\theta(\mathcal{N}) \geq \eta'(w, c, \sigma) > 0$ .*

**Proof.** The proof consists of transforming this MIMO network into a SISO network and then applying the previous theorem. As before, we can assume that all buffer sizes are one in  $\mathcal{N}$  as larger buffers only increase throughput.

Let  $V_s \subseteq V$  and  $V_d \subseteq V$  denote the sets of sources and sinks respectively. We construct a new FJQN/B (see, e.g. Figure 8),  $\mathcal{N}^t$  as follows.  $V^t = V \cup \{s, d\}$ ,  $E^t = E \cup \{(s, v) : v \in V_s\} \cup \{(v, d) : v \in V_d\}$ . We set the service times of these two servers to zero,  $\sigma_{i,n}^t = 0, i \in \{s, d\}$ . All of the buffers are set to size one. We claim without proof that

$$R_{i,n}^t \geq R_{i,n}, \quad n = 1, \dots; \quad v \in R_i$$

This is because a source cannot begin to emit a new customer until all other sources have emitted a customer with the same index and a sink cannot emit a customer until all of the other sinks have emitted a customer with the same index. ■

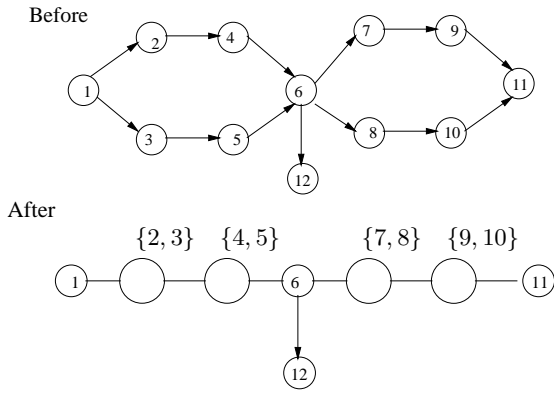


**Figure 8: Transform a MIMO network into a SISO network.**

We now consider the case where  $\mathcal{N}$  is an interconnection of maximal SC FJQN/Bs. Such a network can be visualized as a tree within which nodes may correspond to individual servers or to SC subgraphs. If one were to replace every maximal SC component with a single node, the resulting graph would be a tree.

**THEOREM 24.** *Under Assumption 1 and assume further the service times are light tailed, a "tree" consisting of  $w$ -wide,  $c$ -balanced maximal SC components and individual servers with degree (in-degree + out-degree) less than or equal to  $k$  has a throughput  $\theta(\mathcal{N}) \geq \eta''(k, c, w, \sigma) > 0$ .*

**Proof.** We will construct a FJQN/B  $\mathcal{N}^t$  whose graph  $G^t$  is a tree with service times bounded from above by  $\sigma^{(wc)}$ .



**Figure 9: Transform a MIMO network with multiple maximal SC components into a tree.**

To do so, we consider each maximal SC component and transform it into a tandem network using the construction described in the previous two theorems. Label the maximal SC components  $j = 1, \dots, M$ . Let  $R_{j,i}$  denote the set of nodes contained in the  $i$ -th node of the reduced tandem network associated with component  $j$ . For every node  $v \in R_{j,i}$  such that there exists a predecessor  $k$  not in component  $j$ , add an edge from  $k$  to the  $i$ -th node in the reduced tandem network. If  $k$  has been aggregated into some other node, say  $R_{j',i'}$  in tandem network  $j'$ , then the edge is added from  $R_{j',i'}$  to  $R_{j,i}$ . Likewise if there exists a successor,  $k$ , not in component  $j$ , add an edge to that node (or its aggregate node) from the  $i$ -th node in the reduce tandem network.

This construction will produce a FJQN/B consisting of an in tree attached to an out tree. Lemma 20 is crucial here, both to show that the reduction of SC components to tandem networks degrades throughput and to show that the interconnection among these tandem networks as described above also further reduces throughput. Last, we apply Lemma 21 to show that  $\mathcal{N}^t$  has the same throughput as one constructed from  $\mathcal{N}^l$  by reversing all edges within the in-tree (Lemma 21). The theorem then follows from the throughput scalability of tree networks based on Lemma 7. ■

Figure 9 illustrates, for example, how a tree with two  $w$ -wide,  $c$ -balanced maximal SC components can be transformed into a tree.

## 7. CONCLUDING REMARKS

The prevalence of large-scale stream processing networks has placed the network scalability as the central problem for practitioners and system engineers. Most previous studies address the scalability problem from the engineering perspective. There is an urgent need to develop the mathematical foundation for investigating the scalability problem since conducting experimental studies on large-scale networks is practically infeasible.

This paper adapts a mathematical framework and investigates the throughput scalability of large-scale stream processing networks. We model such networks as general fork/join queueing networks with blocking, where the processing speed and buffer space of each node does not grow with the size of the network.

We establish performances bounds on the asymptotic through-

put, from which we derive necessary conditions under which the network stays throughput scalable. We also study in greater detail the special class of series-parallel networks. We present explicit conditions under which the system throughput would scale and under which it wouldn't. We further provide the exact speed of the throughput degradation as the system size grows. These results, from a mathematical perspective, potentially further the understanding of the difficult problem of analyzing large queueing networks with finite buffers, for which there is no general explicit solutions such as the product-form of Jackson networks. We show as an example that, using the duality property, our results in a two-branch series-parallel network can be interpreted in the closed queueing network setting with finite buffers.

Finally, we report positive results for a large class of FJQN/B networks, namely those that are "balanced" and not too "wide". It is shown that throughput of these networks is bounded below by a positive number independent of the size.

Although the positive results currently rely on the 'bounded-width' condition, we believe it is not necessary. We conjecture that our necessary conditions are also sufficient to guarantee scalable throughput. In other words, if the network is well balanced and all nodes have bounded degrees, we conjecture that the system throughput will be bounded below by a positive number regardless of the network size. Some preliminary investigations show that the proof of such sufficiency will be much more involved.

## 8. REFERENCES

- [1] D. J. Abadi et al. The design of the borealis stream processing engine. In *Proc. of CIDR*, pages 277–289, 2005.
- [2] F. Baccelli, A. Chaintreau, Z. Liu, and A. Riabov. The one-to-many tcp overlay: a scalable and reliable multicast architecture. In *Proc. IEEE INFOCOM*, 2005.
- [3] F. Baccelli and D. Hong. Tcp is max-plus linear, and what it tells us on its throughput. In *Proc. of SIGCOMM*, pages 219–230, 2000.
- [4] F. Baccelli and Z. Liu. On the stability condition of a precedence-based queueing discipline. *Adv. Appl. Prob.*, 21:883–887, 1989.
- [5] F. Baccelli and Z. Liu. On the execution of parallel programs on multiprocessor systems—a queueing theory approach. *Journal of the ACM*, 37(2):373–417, 1990.
- [6] F. Baccelli and Z. Liu. Comparison properties of stochastic decision free petri nets. *IEEE Trans. on Automatic Control*, 37:1905–1920, 1992.
- [7] D. Bhattacharyya, D. Towsley, and J. Kurose. The loss path multiplicity problem in multicast congestion control. In *Proc. of IEEE INFOCOM*, 1999.
- [8] A. Chaintreau. *Processes of Interaction in Data Networks*. PhD thesis, Ecole Normale Supérieure and Universit Paris 6, 2006.
- [9] A. Chaintreau, F. Baccelli, and C. Diot. Impact of network delay variations on multicast sessions with tcp-like congestion control. pages 1133–1142, 2001.
- [10] Y. Chawathe, S. McCanne, and E. Brewer. Rmx: Reliable multicast in heterogeneous networks. In *Proc. of IEEE INFOCOM*, 2000.
- [11] L. Chen, K. Reddy, and G. Agrawal. Gates: A

- gridbased middleware for processing distributed data streams. In *Proc. of HPDC*, 2004.
- [12] M. Cherniack et al. Scalable distributed stream processing. In *Proc. of CIDR*, 2003.
- [13] Y. Dallery, Z. Liu, and D. Towsley. Equivalence, reversibility, symmetry and concavity properties in fork/join queueing networks with blocking. *Journal of the ACM*, 41:903–943, 1994.
- [14] Y. Dallery, Z. Liu, and D. Towsley. Properties of fork/join queueing networks with blocking under various operating mechanisms. *IEEE Transactions on Robotics and Automation*, 13:503–518, 1997.
- [15] A. Das, J. Gehrke, and M. Riedewald. Approximate join processing over data streams. In *Proc. ACM SIGMOD*, 2003.
- [16] Z. Fu, X. Meng, and S. Lu. How bad tcp can perform in mobile adhoc networks. *IEEE Symposium on Computers and Communications*, 2002.
- [17] Z. Fu, X. Meng, and S. Lu. A transport protocol for supporting multimedia streaming in mobile ad hoc networks. *IEEE journal on selected areas in communications*, 21(10):1615–1626, 2004.
- [18] J. Galambos. *The Asymptotic Theory of Extreme Order Statistics*. Wiley, NY, 1978.
- [19] P. Gibbons, B. Karp, Y. Ke, S. Nath, and S. Seshan. Irisnet: An architecture for a world-wide sensor web. *IEEE Pervasive Computing*, 2(4), 2003.
- [20] P. Hsiao, H. Kung, and K. Tan. Active delay control for tcp. In *IEEE Globecom*, 2001.
- [21] Y. Huang, W. Gong, and D. Towsley. Application layer relays for wireless 802.11 mesh networks. In *Proc. of IEEE workshop WiMesh*, pages 81–90, 2006.
- [22] N. Jain, L. Amini, H. Andrade, R. King, Y. Park, P. Selo, and C. Venkatramani. Design, implementation, and evaluation of the linear road benchmark on the stream processing core. In *Proc. of SIGMOD*, pages 431–442, 2006.
- [23] J. Jannotti, D. Gifford, K. Johnson, M. Kaashoek, and J. O. JR. Overcast: Reliable multicasting with an overlay network. In *Proc. 4th USENIX OSDI*, pages 197–212, 2000.
- [24] P. Jelenkovic, P. Momcilovic, and M. Squillante. Buffer scalability of wireless networks. In *Proc. of IEEE Infocom*, 2006.
- [25] G. Kwon and J. Byers. Roma: Reliable overlay multicast with loosely coupled tcp connections. In *Proc. of IEEE INFOCOM*, 2004.
- [26] J. Martin. Large tandem queueing networks with blocking. *QUESTA*, 41:45–72, 2002.
- [27] M. D. Mascolo, R. David, and Y. Dallery. Modeling and analysis of assembly systems with unreliable machines and finite buffers. 23(4):315–330, 1991.
- [28] P. Mehra and A. Zakhor. Tcp-based video streaming using receiver-driven bandwidth sharing. In *Int'l Packet Video Workshop*, 2003.
- [29] D. V. Schuehler. *TCP Stream Processing at Gigabit Line Rates*. PhD thesis, Department of Computer Science and Engineering, Washington University, 2004.
- [30] J. A. Sharp. *Data Flow Computing*. Ablex Publication Corp., 1991.
- [31] D. Simchi-Levi, P. Kaminsky, and E. Simchi-Levi. *Designing and Managing the Supply Chain*. McGraw-Hill/Irwin, second edition, 2002.
- [32] D. Stoyan. *Comparison Methods for Queues and Other Stochastic Processes*, 1983.
- [33] B. Wang, J. Kurose, P. Shenoy, and D. Towsley. Multimedia streaming via tcp: An analytic performance study. In *Proc. of ACM Multimedia*, 2004.

## Appendix

### 8.1 Proof of Lemma 2 and Lemma 3.

The vector  $x \in \mathbb{R}^n$  is said to be *majorized* by the vector  $y \in \mathbb{R}^n$ , written  $x \prec y$ , iff  $\sum_{i=1}^k x_{[i]} \leq \sum_{i=1}^k y_{[i]}$ ,  $k = 1, \dots, n-1$ , and  $\sum_{i=1}^n x_{[i]} = \sum_{i=1}^n y_{[i]}$ , where  $x_{[i]}$  (resp.  $y_{[i]}$ ) denotes the  $i$ -th largest component of  $x$  (resp.  $y$ ).

A function  $\phi : \mathbb{R}^n \mapsto \mathbb{R}$  is called *Schur convex* if  $\phi(x) \leq \phi(y)$  for all  $x \prec y$ . A function that is symmetric and convex in each pair of arguments is known to be Schur convex.

**Proof of Lemma 2.** It suffices to prove the lemma for  $n = 2$ . (The same argument can be repeated for higher dimensions). Denote  $Y = \frac{X_1 + X_2}{2}$ . Then  $(Y, Y) \prec (X_1, X_2)$ . For any convex function  $f : \mathbb{R} \mapsto \mathbb{R}$ , it is easily checked that  $\phi(x, y) = f(x) + f(y)$  is a Schur convex function, which preserves the  $\prec$ -ordering. Hence,

$$E[f(Y)] = \frac{1}{2}E[\phi(Y, Y)] \leq \frac{1}{2}E[\phi(X_1, X_2)] = E[f(X_0)].$$

We then have  $Y \leq_{cx} X_0$ . ■

**Proof of Lemma 3.** Let  $R_n(\mathcal{N}; \sigma)$  (resp.  $R_n(\mathcal{N}; \sigma')$ ) denote the service completion time of job  $n$  in the FJQN/B  $\mathcal{N}$  with service time sequence  $\{\vec{\sigma}_n\}_n$  (resp.  $\{\vec{\sigma}'_n\}_n$ ).

For a FJQN/B  $\mathcal{N} = (V, E, \mathbf{B})$  with initial marking  $\mathbf{M}$ , [14] defines (see Definition 5.1 and Definition 5.2 of [14]) a precedence graph  $\mathcal{G}_{\mathcal{N}} = (\mathcal{V}, \mathcal{E})$  associated with FJQN/B  $\mathcal{N}$ , where

$$\begin{aligned} \mathcal{V} &= \{(i, n) | n \geq 1, i \in V\}, \\ \mathcal{E} &= \{(i, n) \rightarrow (j, m) | (i, n), (j, m) \in \mathcal{V}, (i, n) \prec_{\mathcal{N}} (j, m)\}, \end{aligned}$$

and the relation  $(i, n) \prec_{\mathcal{N}} (j, m)$  holds iff one of the following relations is satisfied:  $n = m - M_{i,j}$ ,  $i \in p(j)$ ; or  $n = m - 1$ ,  $i = j$ ; or  $n = m - (B_{j,i} - M_{j,i})$ ,  $i \in s(j)$ . Note that such a precedence graph depends only on the network topology  $\mathcal{N}$ , the buffer sizes  $B$ , and the initial marking  $\mathbf{M}$ .

Let  $\mathcal{P}$  denote the set of all paths in  $\mathcal{G}_{\mathcal{N}}$ . Define

$$S_{1,n}(\sigma) = \max_{p = ((i_1, n_1) \rightarrow \dots \rightarrow (i_k, n_k)) \in \mathcal{P}: n_1 = 1, n_k = n} \sum_{h=1}^k \sigma_{i_h, n_h}. \quad (16)$$

Based on Theorem 5.6 in [14], there exists a constant  $\mu_{\sigma}$  such that

$$\mu_{\sigma} = \lim_{n \rightarrow \infty} \frac{R_n(\mathcal{N}; \sigma)}{n} = \lim_{n \rightarrow \infty} \frac{S_{1,n}(\sigma)}{n}, \quad \text{almost surely.} \quad (17)$$

Under service time sequence  $\{\vec{\sigma}'_n\}_n$ , one can similarly define  $S_{1,n}(\sigma')$  and  $\mu_{\sigma'}$ . Since  $\vec{\sigma}_n \stackrel{d}{=} \gamma \cdot \vec{\sigma}'_n$ , it follows easily from (16) that

$$S_{1,n}(\sigma) \stackrel{d}{=} \gamma \cdot S_{1,n}(\sigma'). \quad (18)$$

Combine (4), (17) and (18), then yields  $\theta(\mathcal{N}; \sigma) = \frac{1}{\gamma} \theta(\mathcal{N}; \sigma')$ . ■