

Control of Fork-Join Networks in Heavy Traffic

A. Zviran¹ R. Atar² A. Mandelbaum¹

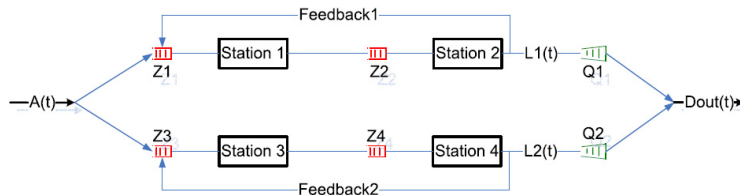
¹Faculty of Industrial Engineering and Management
Technion

²Faculty of Electrical Engineering
Technion

Allerton Conference, 2012

Introduction

Our Network Model



- A **Fork-Join Network** (FJN) is a natural model for a queuing system in which customers are processed both sequentially and in parallel.
- Our generalized fork-join structure allows probabilistic feedback.
- The identity of the job being processed at time t by station j , is regarded as the **scheduling control**.

Main Idea and Motivation

Parallel processing systems are commonly encountered in many human ventures.

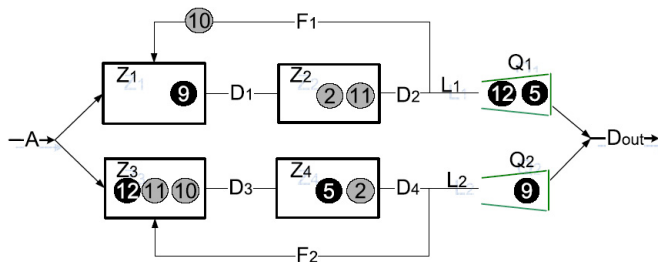
Main idea: A simple close-loop scheduling control is used to increase throughput and reduce synchronization delays.

Motivation: Parallel Processing Application

- Parallel communication networks.
- Data storage allocation.
- Large scale parallel computing.
- Multi-Project scheduling.
- Health-care systems (service networks).

Synchronization Constraints? (FJN)

In our model, **tasks** are associated uniquely with **customers**. They are hence **non-exchangeable** in the sense that one can not join tasks associated with different customers.

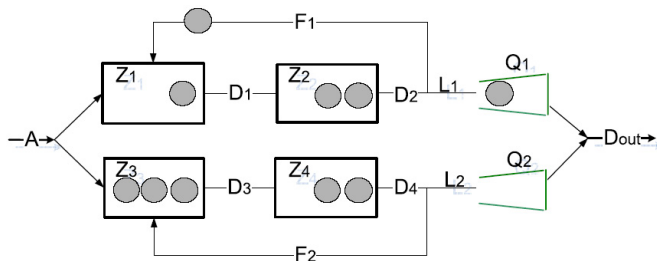


The markovian feedback reshuffles the order of the departing tasks, causing delays in the synchronization queues.

Conclusion- Customers' **disorder** increase server Idle-times in the join nodes and decrease throughput.

Synchronization Constraints? (Assembly Network)

This is in contrast to **Assembly network** in which customers are **exchangeable**, thus join and depart the system regardless of order.



An (exchangeable) assembly network can thus be characterized by the following **Complementarity Condition**

$$Q_1(t) \wedge Q_2(t) = 0, \quad \forall t \geq 0$$

Methodology: Asymptotic Analysis

We shall work in the **conventional Heavy-Traffic regime**.

The precise formulation of Heavy-Traffic limits requires the construction of a "sequence of systems", indexed by $n = 1, 2, \dots$

Assume that the following relations hold

- Average arrival rate: $\lambda^n = \lambda \cdot n + \hat{\lambda} \cdot \sqrt{n} + o(\sqrt{n})$.
- Average service rates: $\mu_j^n = \mu_j \cdot n + \hat{\mu}_j \cdot \sqrt{n} + o(\sqrt{n})$.
- Heavy Traffic Condition - Define the traffic intensity at station j to be ρ_j^n , it is assumed that there exists a deterministic number $-\infty < \theta_j < \infty$, such that $\sqrt{n} \cdot (\rho_j^n - 1) \rightarrow_n \theta_j$, as $n \rightarrow \infty$, for each station j .

Notation - Throughout the presentation we shall use the scaling

$$\hat{Q}_i^n(t) = \frac{Q_i^n(t)}{\sqrt{n}}.$$

Control Problem Formulation

Definition of Optimality

Throughput Optimality is defined in terms of maximal achievable number of departures over a finite time-horizon, or more precisely,

Definition

- **Exact Optimality:** Control $\gamma \in \mathbb{A}$ is optimal if, for all $T > 0$, γ attains $\text{esssup}_{\alpha \in \mathbb{A}}(D_{out}^{\alpha}(T))$.
- **Asymptotic Optimality:** Control $\gamma \in \mathbb{A}$ is asymptotically optimal if for any other control $\alpha \in \mathbb{A}$ and for all $T > 0$,

$$\hat{D}_{out}^{n,\gamma}(T) \geq \hat{D}_{out}^{n,\alpha}(T) - \epsilon_n(T), \text{ with } \epsilon_n(T) \rightarrow 0,$$

where the convergence of $\epsilon_n(\cdot)$ is u.o.c, in probability.

Optimality Criteria

Heuristically, the optimal performance (maximal throughput) is that of a corresponding assembly network, with exchangeable tasks. Indeed, the following **Complementarity Conditions** are sufficient for optimality,

Proposition

Each of the following conditions implies its corresponding definition:

- *Exact Optimality:* $Q_1(T) \wedge Q_2(T) = 0, \forall T > 0;$
- *Asymptotic Optimality:* $\hat{Q}_1^n(\cdot) \wedge \hat{Q}_2^n(\cdot) \xrightarrow{P_n} 0$, where $\xrightarrow{P_n}$ denotes convergence u.o.c., in probability.

Note: Our network model is the simplest settings, in which **First-Come-First-Serve (FCFS)** control is neither optimal nor asymptotically optimal.

Main Result:

Cronyism- or γ -control

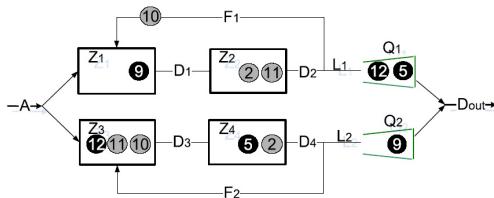
Control Policy

Proposed Control (referred to as Cronyism- or γ -control)

Within each route, assign preemptive priority to tasks of customers whose service was completed in the other route.

- **LP (Low Priority)**: customers whose service is still incomplete in both routes; e.g., gray customers.
- **HP (High Priority)**: customers whose service were completed in one of the routes but is still incomplete in the other; e.g., black customers.

Assume FCFS within each class, which now fully characterizes the control.



Asymptotic Optimality

Theorem (Asymptotic Optimality Theorem)

The scaled process $\hat{Z}_{1,2}^{n,H}(t) \wedge \hat{Z}_{3,4}^{n,H}(t)$ converges u.o.c to 0, in probability



where

$\hat{Z}_{1,2}^{n,H}$ correspond with upper route HP scaled queue length process, and

$\hat{Z}_{3,4}^{n,H}$ correspond with lower route HP scaled queue length process.

But since

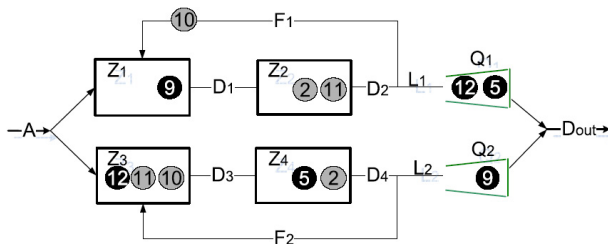
$$\hat{Z}_{1,2}^{n,H}(t) = \hat{Q}_2^n(t) \quad \text{and} \quad \hat{Z}_{3,4}^{n,H}(t) = \hat{Q}_1^n(t),$$

we now conclude that

The scaled process $\hat{Q}_1^n(\cdot) \wedge \hat{Q}_2^n(\cdot)$ converges u.o.c to 0, in probability.

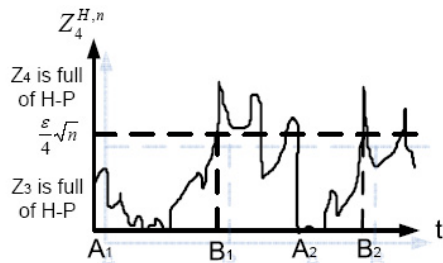
The Technical Challenge In The Proof (1)

- A central part of the proof is to establish tight estimates on HP and LP processes.
- However, our asymptotically optimal γ -control creates a volatile environment of priority switches (LP to HP).
- This renders challenging the characterization of the HP "Birth" processes, indeed their precise analysis would entail tracking the precise station where each task is located.



The Technical Challenge In The Proof (2)

- Instead, rather than making an attempt to characterize these **HP "Birth" processes**, our approach is to develop estimates that are **uniform** over all birth processes.
- A central ingredient in the proof includes the deduction of tightness for the number and length of HP queue **down-crossings**.



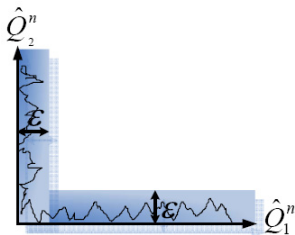
System Dynamics in Heavy Traffic (1)

Recall that $\hat{Z}_{1,2}^{n,H}(t) = \hat{Q}_2^n(t)$ and $\hat{Z}_{3,4}^{n,H}(t) = \hat{Q}_1^n(t)$. We showed that

$\hat{Q}_1^n(t) \wedge \hat{Q}_2^n(t)$ converges u.o.c to 0, in probability.

Corollary (State-space Collapse of Synchronization Queues.)

The relation $\hat{Q}_1^n \wedge \hat{Q}_2^n \stackrel{p}{\approx} 0$ implies that the 2-dimensional stochastic process $\hat{Q}_1^n, \hat{Q}_2^n$ collapses to 1-dimension.



System Dynamics in Heavy Traffic (2)

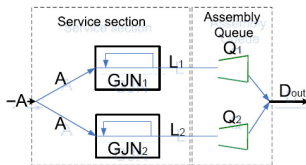
Note that under the γ -control the following holds true:

$D_{out}^{n,\gamma}(\cdot)$ converges u.o.c to $L_1^n(\cdot) \wedge L_2^n(\cdot)$, in probability.

Corollary (Asymptotic Exchangeability.)

In heavy-traffic, applying γ -control to our FJN yields a throughput process D_{out} that has approximately the same distribution as that of an assembly network under FCFS control.

Note: Our network model can be regarded as a special case of a model with **several parallel Jackson networks**.



Final Remarks and Extensions

Conclusions

- We introduced a natural concept of optimality for fork-join networks with **non-exchangeable** customers.
- We proposed a simple closed-loop control, named the **γ -control**, and proves asymptotic throughput optimality.
- **Asymptotic equivalence** appears, in heavy-traffic, between our fork-join network under γ -control and a corresponding **assembly network** under FCFS.
- The distribution of $D_{out} = L_1 \wedge L_2$ is thus **tractable**, in principle, following from the joint distribution of exogenous output processes from a **Generalized Jackson Network**.

Generalization (1)

Note that both the exact and asymptotic conditions may be generalized to any number of parallel processing routes. For M processing routes:

Proposition

Equivalent conditions:

- **Exact Optimality:** $\bigwedge_{i \in \{1, \dots, M\}} (Q_i(T)) = 0$, a.s., for any fixed T ;
- **Asymptotic Optimality:** $\bigwedge_{i \in \{1, \dots, M\}} (\hat{Q}_i^n(\cdot)) \xrightarrow{P} 0$.

According to the following relations

- $M \cdot N(t) = \sum_j (Z_j(t)) + \sum_{i=1}^M (Q_i(t));$
- $\sum_{i=1}^M Q_i(t) = \sum_{i=1}^M (L_i(t) - \bigwedge_{i \in \{1, \dots, M\}} (L_i(t))) + M \cdot \bigwedge_{i \in \{1, \dots, M\}} (Q_i(t));$

Generalization (2)

γ -control for M processing routes

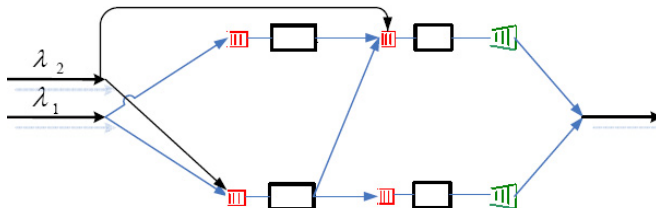
Within each route, assign preemptive priority to tasks of customers whose service was completed in all other routes.

- **LP (Low Priority)**: customers whose service is still incomplete in more than one route.
- **HP (High Priority)**: customers whose service were completed in all other routes but is still incomplete in one route.

Assume FCFS within each class, which now fully characterizes the control.

Extension 1: Multi-Type Model

Consider an **Heterogeneous customer population**, such that different customers may have different precedence constraints, interarrival time distributions and service time distributions, e.g.



Within each route, assign preemptive priority to tasks of customers whose service was completed in the other route. Define **$c\mu$ priority policy** within each class.

Will this modified γ -control be asymptotically optimal for such model?

Extension 2: Halfin-Whitt Regime (QED) Analysis

- In the QED regime, one increases the number of servers with the rate of N , where $N \rightarrow \infty$.
- In this setting, due to a high level of parallel processing, the phenomena of customer overtaking becomes both uncontrollable and non-negligible, in contrast to multi-servers in conventional heavy-traffic.

This gives rise to the following questions:

- is there a control under which Fork-Join and assembly networks are asymptotic equivalent?
- does extremely large number of servers per station presents a disadvantage in parallel processing systems?

Thanks . . .