Systems with unlimited supply of work: MCQN with infinite virtual buffers A Push Pull multiclass system

Gideon Weiss

University of Haifa

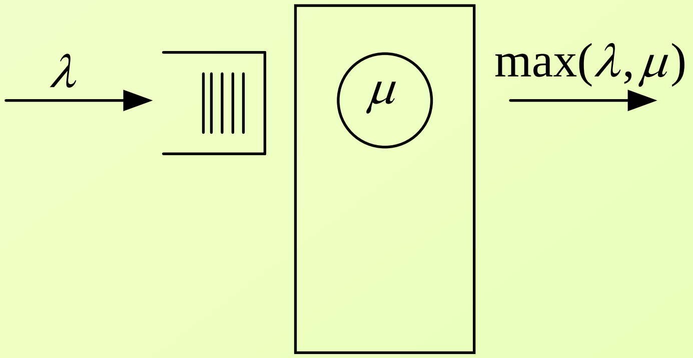
Joint work with students:

Anat (Anastasia) Kopzon

Yoni Nazarathy

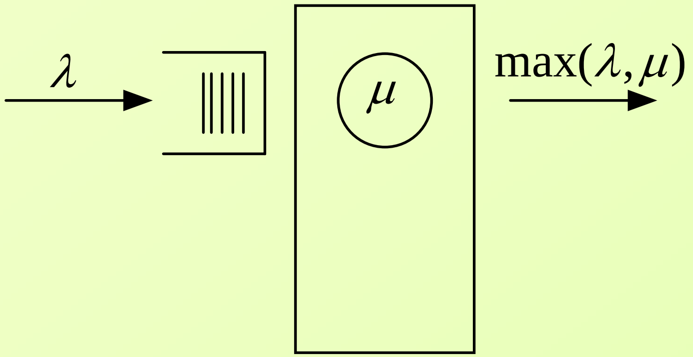
Queue vs Manufacturing machine:

Single server queue

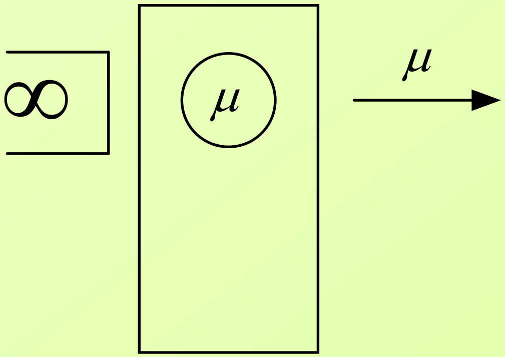


Queue vs Manufacturing machine:

Single server queue

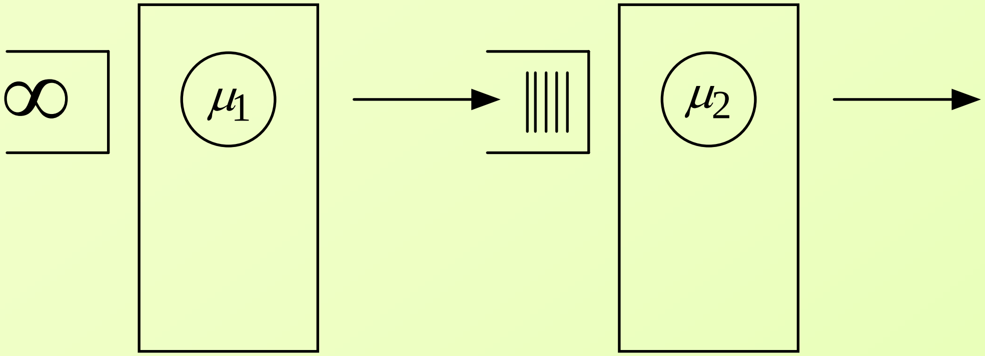


Machine with controlled input



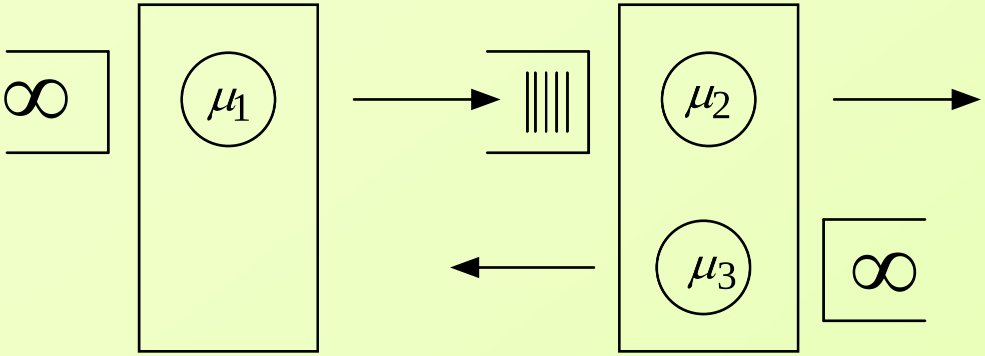
Infinite supply of work - infinite virtual buffers

A tandem of queues



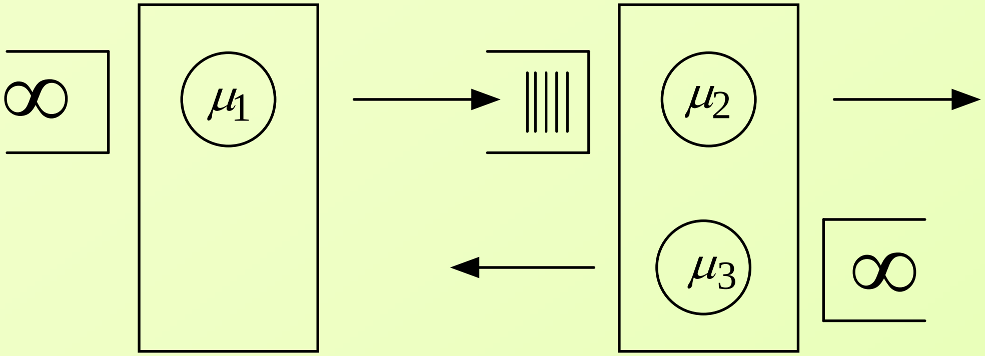
Infinite supply of work - infinite virtual buffers

A tandem of queues

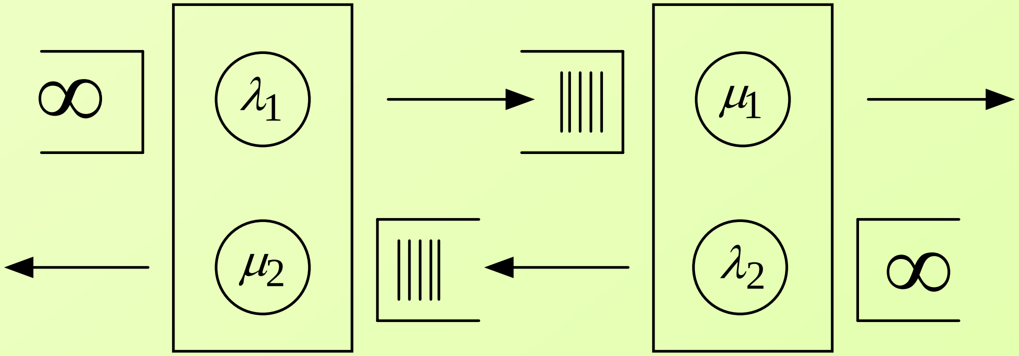


Infinite supply of work - infinite virtual buffers

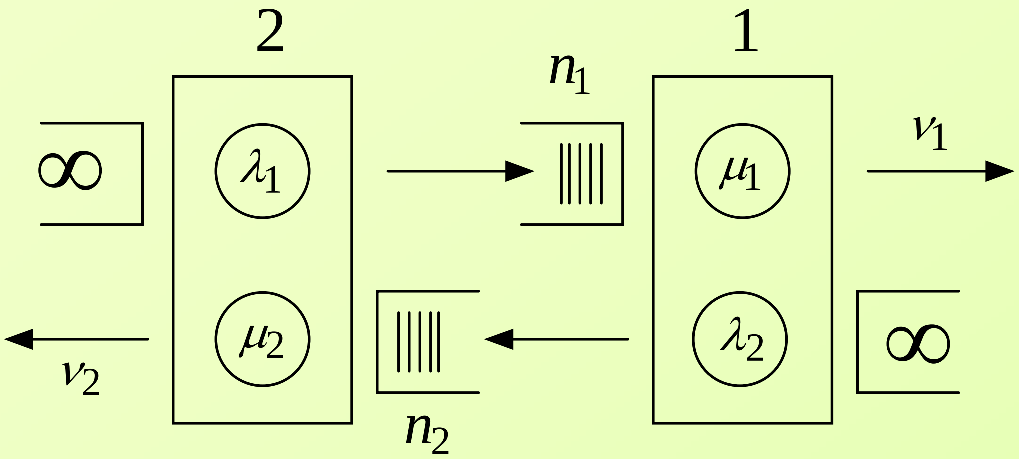
A tandem of queues



The push pull system



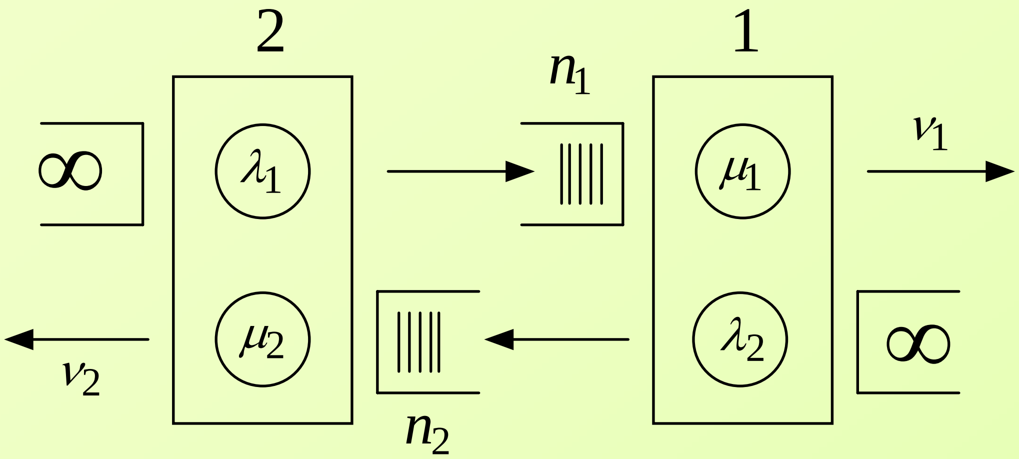
Balanced full utilization



Both machines work all time and no flow accumulates implies:

$$v_1 = \alpha_1\mu_1 = (1-\alpha_2)\lambda_1$$
$$v_2 = \alpha_2\mu_2 = (1-\alpha_1)\lambda_2$$

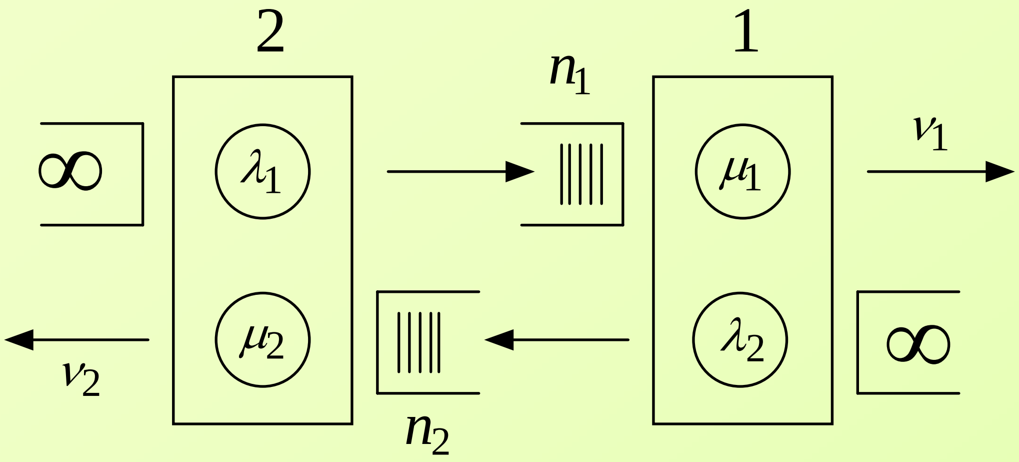
Balanced full utilization



$$v_1 = \frac{\lambda_1\mu_1(\mu_2 - \lambda_2)}{\mu_1\mu_2 - \lambda_1\lambda_2}$$

$$v_2 = \frac{\lambda_2\mu_2(\mu_1 - \lambda_1)}{\mu_1\mu_2 - \lambda_1\lambda_2}$$

Balanced full utilization

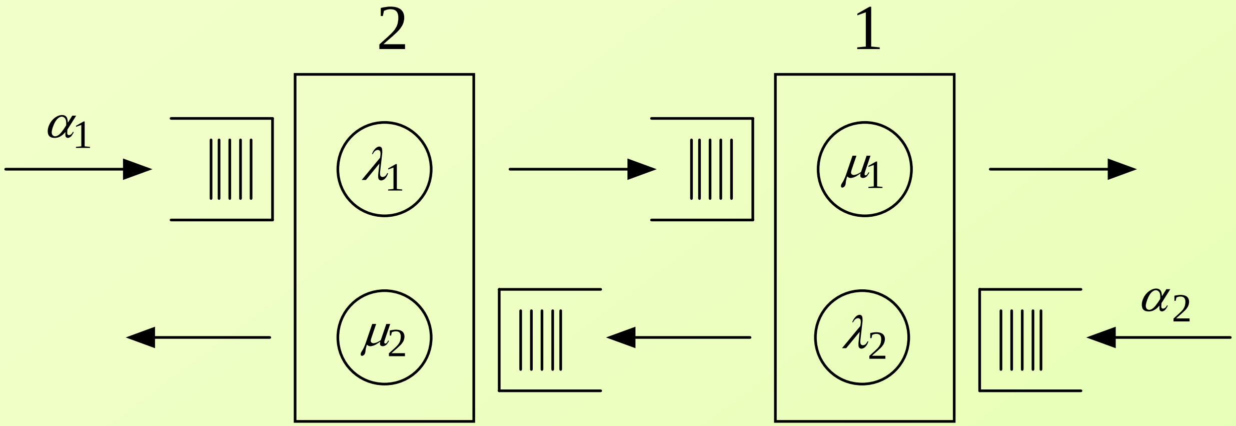


$$v_1 = \frac{\lambda_1 \mu_1 (\mu_2 - \lambda_2)}{\mu_1 \mu_2 - \lambda_1 \lambda_2}$$

$$v_2 = \frac{\lambda_2 \mu_2 (\mu_1 - \lambda_1)}{\mu_1 \mu_2 - \lambda_1 \lambda_2}$$

How does it behave?

The Rybko Stolyar network

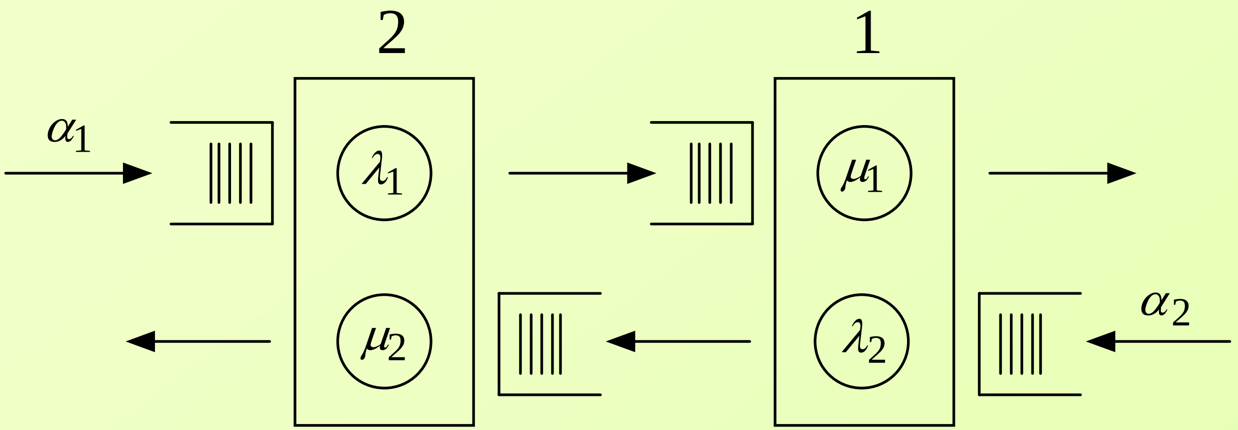


Traffic intensity/offered load

$$\rho_1 = \frac{\alpha_1}{\mu_1} + \frac{\alpha_2}{\lambda_2}$$

$$\rho_2 = \frac{\alpha_2}{\mu_2} + \frac{\alpha_1}{\lambda_1}$$

The Rybko Stolyar network



Traffic intensity

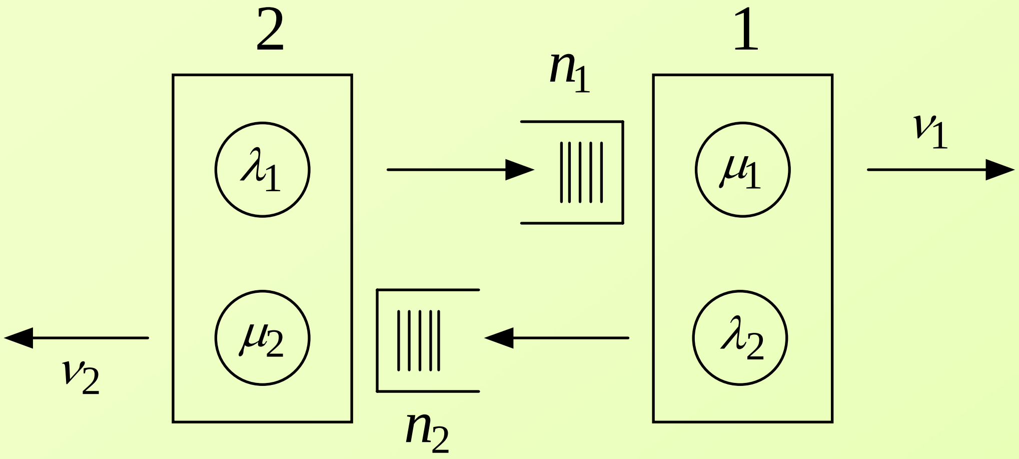
$$\rho_1 = \frac{\alpha_1}{\mu_1} + \frac{\alpha_2}{\lambda_2}$$

$$\rho_2 = \frac{\alpha_2}{\mu_2} + \frac{\alpha_1}{\lambda_1}$$

Heavy traffic: $\alpha_1 \nearrow, \alpha_2 \nearrow \Rightarrow \rho_1 \nearrow, \rho_2 \nearrow$

Balanced heavy traffic: $\alpha_1 \to \nu_1, \quad \alpha_2 \to \nu_2$

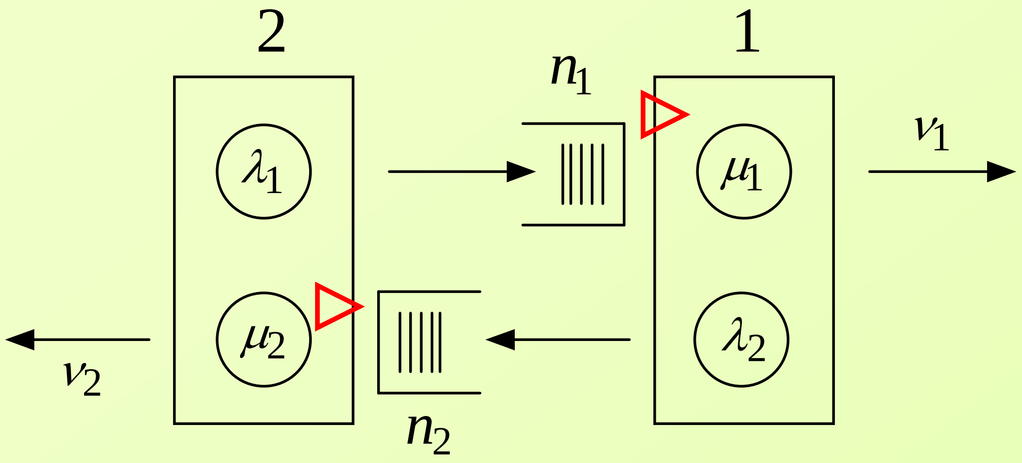
Push pull system - inherently stable case



$\lambda_1 < \mu_1$

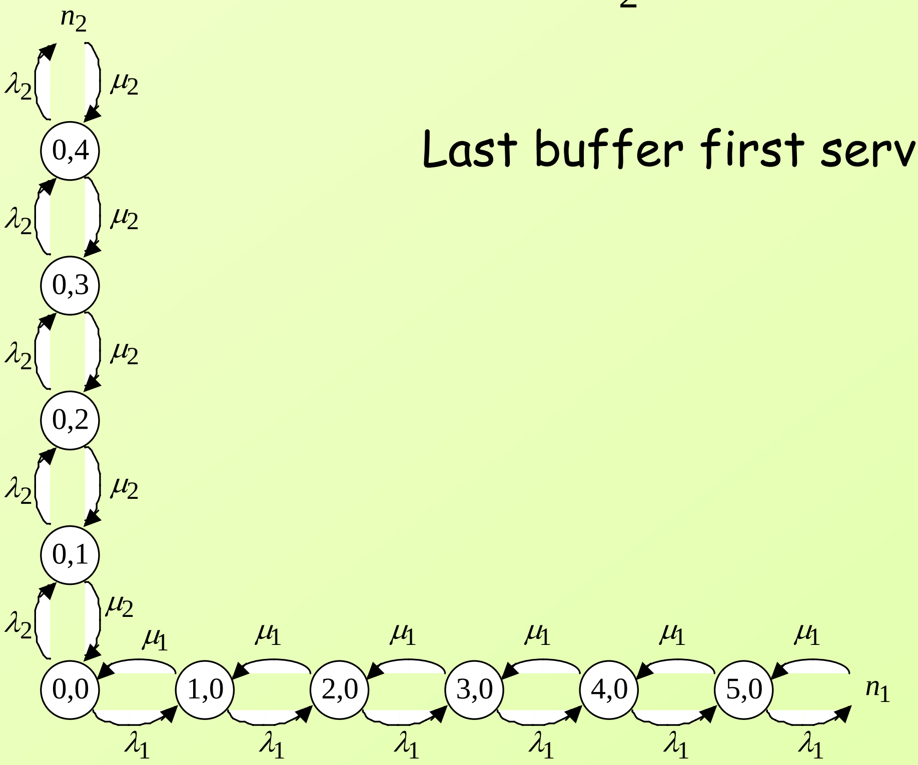
$\lambda_2 < \mu_2$

Push pull system - inherently stable case



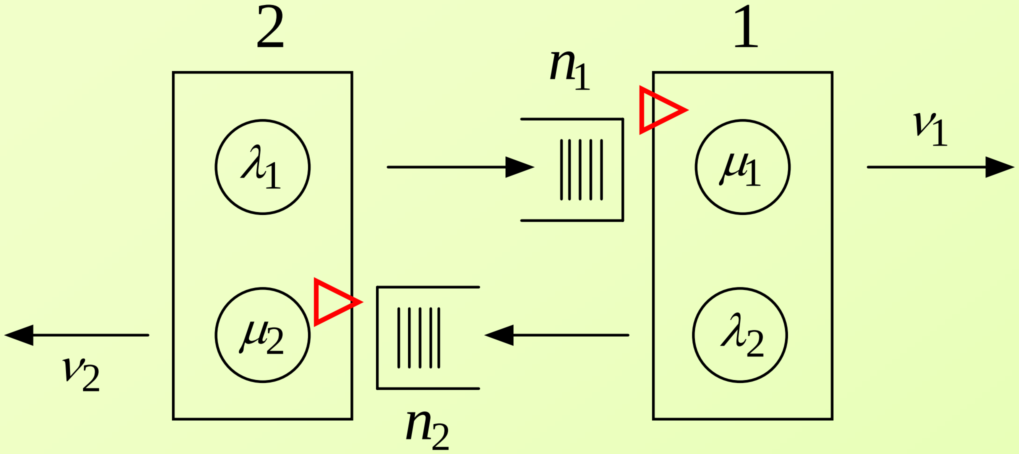
$\lambda_1 < \mu_1$

$\lambda_2 < \mu_2$



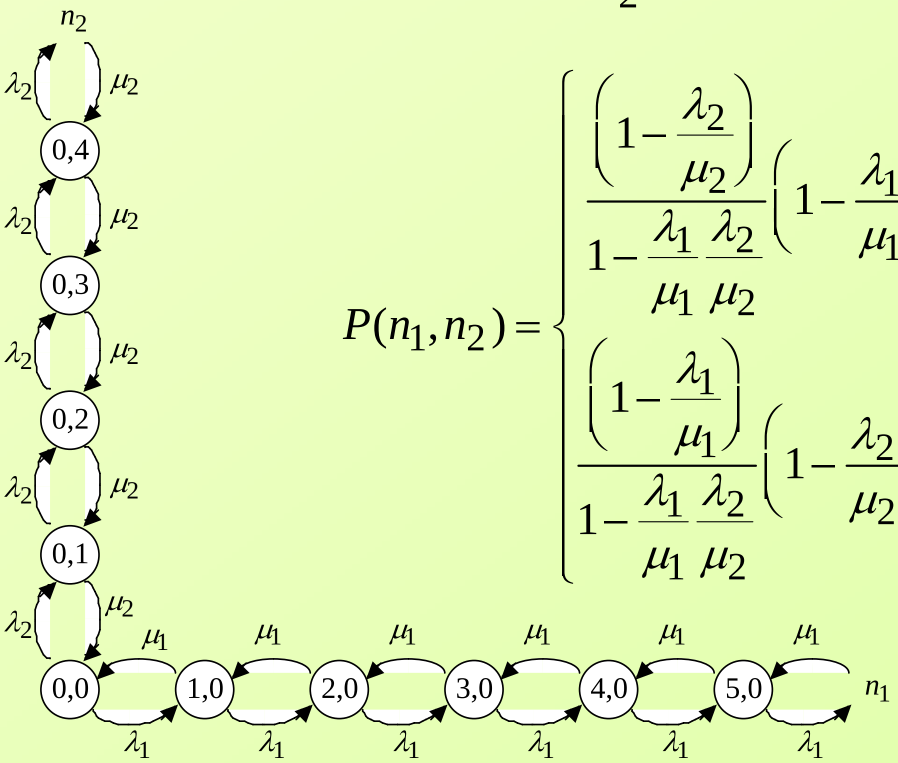
Last buffer first serve, priority to pull over push

Push pull system - inherently stable case



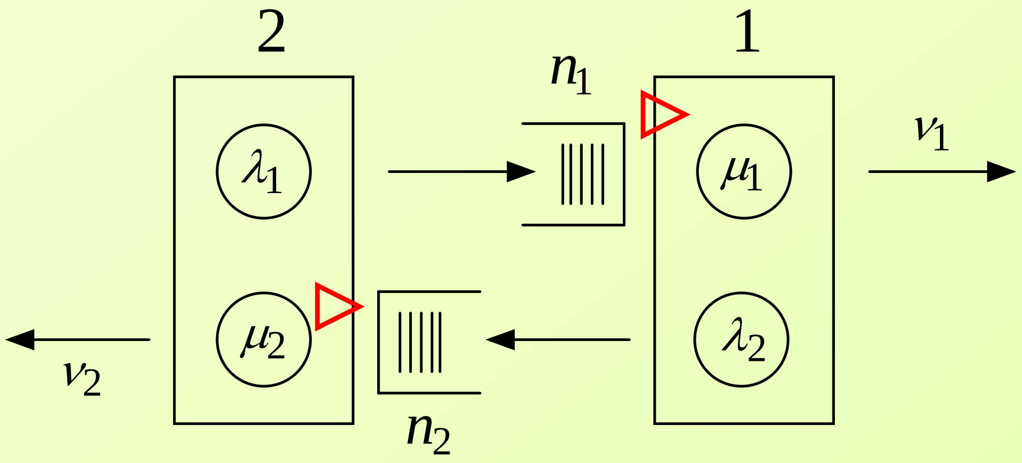
$\lambda_1 < \mu_1$

$\lambda_2 < \mu_2$



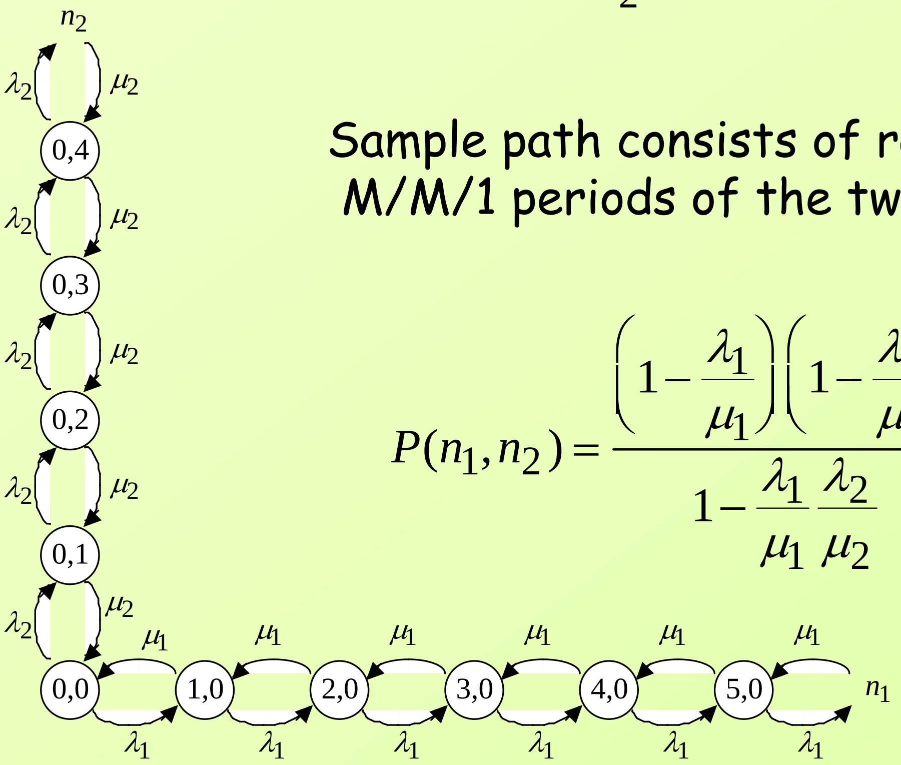
$$P(n_1,n_2)=\begin{cases} \dfrac{\left(1-\dfrac{\lambda_2}{\mu_2}\right)}{1-\dfrac{\lambda_1}{\mu_1}\dfrac{\lambda_2}{\mu_2}}\left(1-\dfrac{\lambda_1}{\mu_1}\right)\left(\dfrac{\lambda_1}{\mu_1}\right)^{n_1}, & n_2=0 \\ \dfrac{\left(1-\dfrac{\lambda_1}{\mu_1}\right)}{1-\dfrac{\lambda_1}{\mu_1}\dfrac{\lambda_2}{\mu_2}}\left(1-\dfrac{\lambda_2}{\mu_2}\right)\left(\dfrac{\lambda_2}{\mu_2}\right)^{n_2}, & n_1=0 \end{cases}$$

Push pull system - inherently stable case



$\lambda_1 < \mu_1$

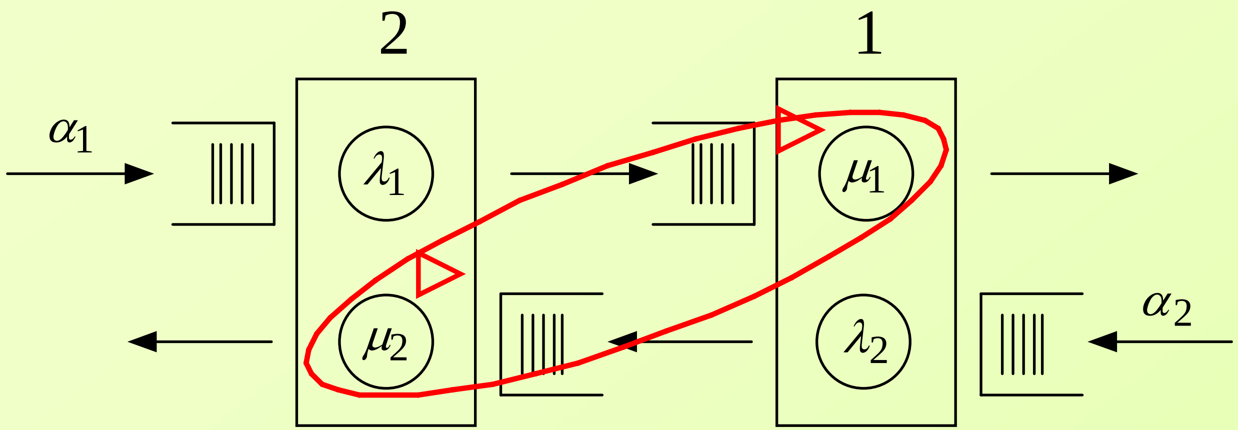
$\lambda_2 < \mu_2$



Sample path consists of randomly switching between M/M/1 periods of the two streams

$$P(n_1,n_2)=\frac{\left(1-\frac{\lambda_1}{\mu_1}\right)\left(1-\frac{\lambda_2}{\mu_2}\right)}{1-\frac{\lambda_1}{\mu_1}\frac{\lambda_2}{\mu_2}}\left(\frac{\lambda_1}{\mu_1}\right)^{n_1}\left(\frac{\lambda_2}{\mu_2}\right)^{n_2}, \quad n_1\cdot n_2=0$$

Rybko Stolyar network - LBFS virtual machine



Under LBFS the two pulling queues form a virtual machine -
only one works at any time

Conditions for stability (global stability of all work conserving policies)

$$\rho_1 = \frac{\alpha_1}{\mu_1} + \frac{\alpha_2}{\lambda_2}$$

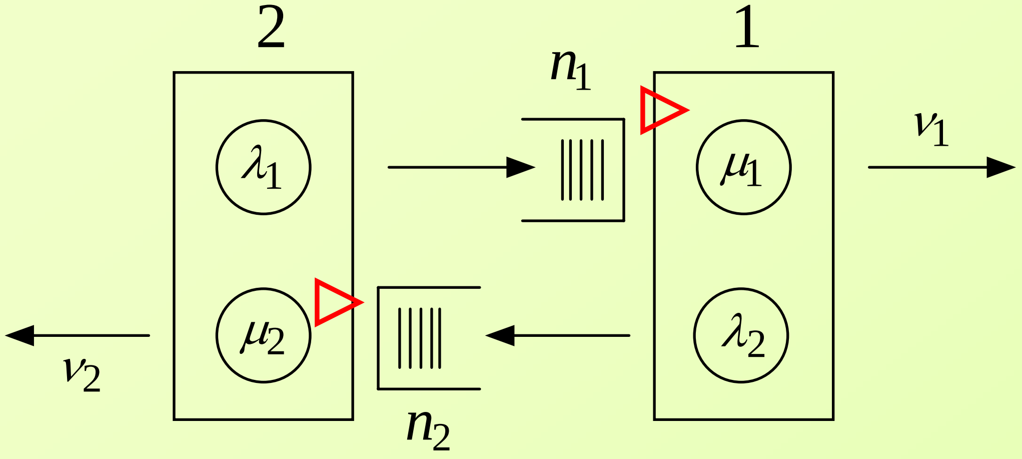
$$\rho_2 = \frac{\alpha_2}{\mu_2} + \frac{\alpha_1}{\lambda_1}$$

$$\text{virtual machine load} = \frac{\alpha_1}{\mu_1} + \frac{\alpha_2}{\mu_2} < 1$$

When $\lambda_i > \mu_i$, we can have $\rho_1, \rho_2 < 1$ but virtual machine load > 1

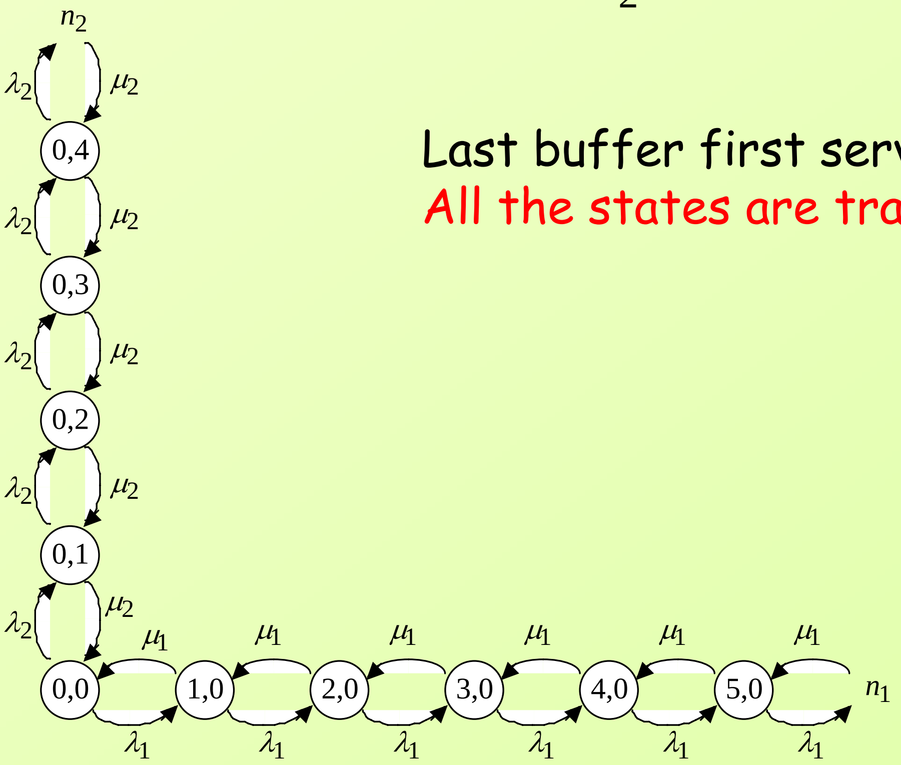
LBFS will be unstable

Push pull system - inherently unstable case



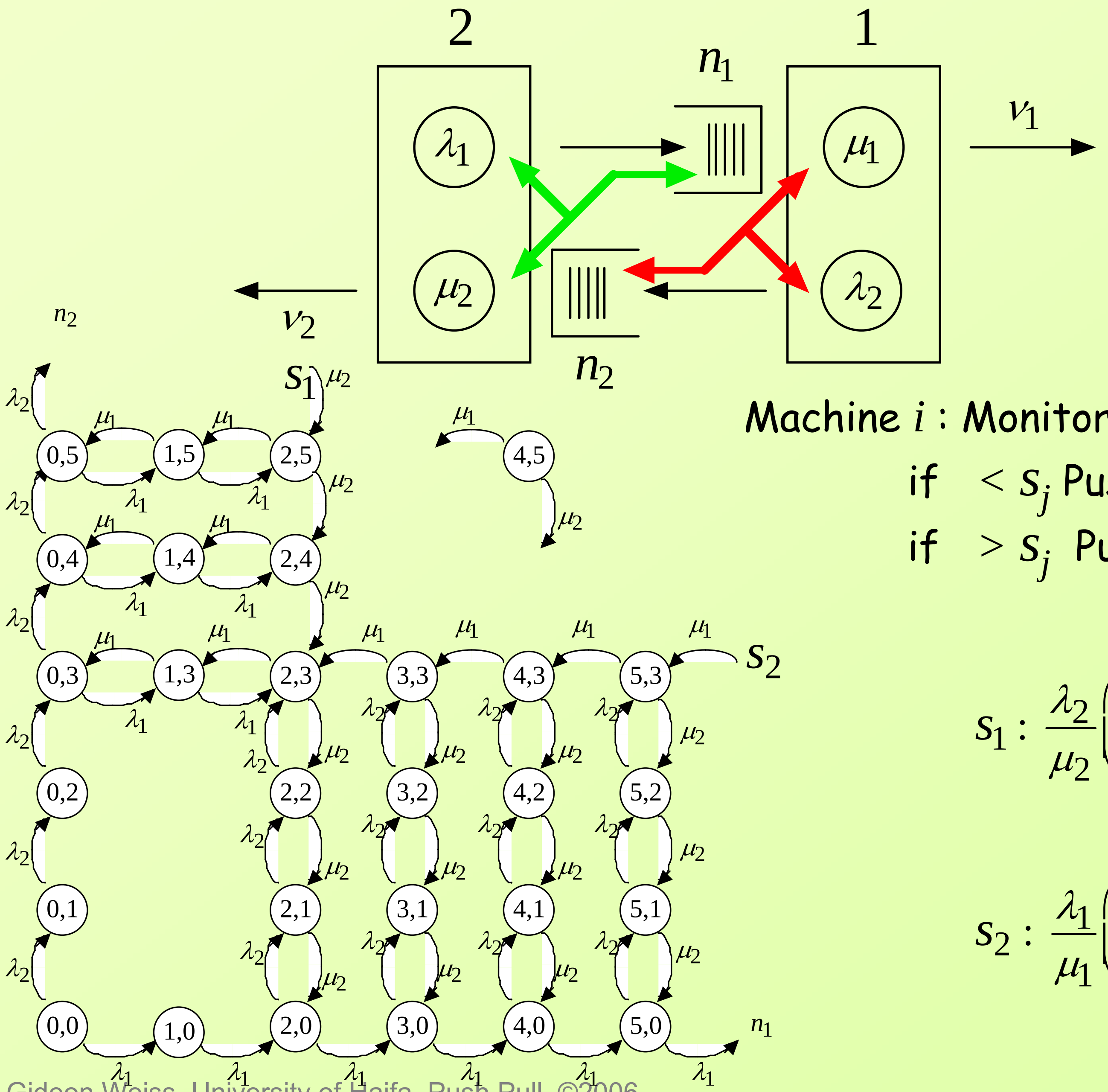
$\lambda_1 > \mu_1$

$\lambda_2 > \mu_2$



Last buffer first serve, priority to pull over push:
All the states are transient

Push pull system - fixed threshold policy



$\lambda_1 > \mu_1$

$\lambda_2 > \mu_2$

Machine i : Monitor queue at Machine j,

if $< s_j$ Push,

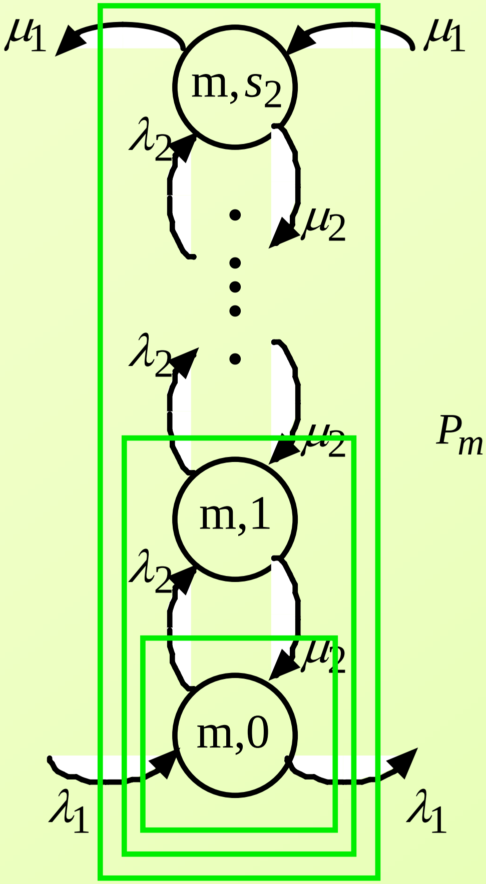
if $> s_j$ Pull.

$$s_1 : \frac{\lambda_2}{\mu_2}\left(\frac{\mu_1}{\lambda_1}\right)^{s_1} < 1$$

$$s_2 : \frac{\lambda_1}{\mu_1}\left(\frac{\mu_2}{\lambda_2}\right)^{s_2} < 1$$

Push pull system - fixed threshold Steady State:

Symmetric streams



$$P_{m,n}=\begin{cases} P_{s,s}\dfrac{\left(\frac{\lambda}{\mu}\right)^n+\frac{\lambda}{\lambda-\mu}\left(\left(\frac{\lambda}{\mu}\right)^n-1\right)}{\left(\frac{\lambda}{\mu}\right)^s+\frac{\lambda}{\lambda-\mu}\left(\left(\frac{\lambda}{\mu}\right)^s-1\right)} & m=s,0\le n\le s \\ P_{s,s}\dfrac{\left[\frac{\lambda}{\mu}+\frac{\lambda}{\lambda-\mu}\left(\left(\frac{\lambda}{\mu}\right)^s-1\right)\right]^{m-s-1}}{\left[\left(\frac{\lambda}{\mu}\right)^s+\frac{\lambda}{\lambda-\mu}\left(\left(\frac{\lambda}{\mu}\right)^s-1\right)\right]^{m-s+1}}\left[2\left(\frac{\lambda}{\mu}\right)^{n+1}+\frac{\lambda}{\lambda-\mu}\left(\left(\frac{\lambda}{\mu}\right)^s-\frac{\lambda}{\mu}\right)\right] & m>s,0\le n\le s \end{cases}$$

Maximum pressure policy

Maximum pressure policy will stabilize any system with offered load <1

Consider MCQN with fluid dynamics described by

$$\frac{d}{dt}Q(t) = \alpha - Ru(t)$$

where R is the input output matrix, $u(t)$ is the machine allocation, and α is the input rate.

The machine allocations (controls) are subject to resource constraints

Max pressure attempts to maximize the gradient of the sum of squares of queue lengths

$$\frac{d}{dt}\sum_k Q_k^2(t) = \frac{d}{dt}Q'(t)Q(t) = 2Q'(t)(\alpha - Ru(t))$$

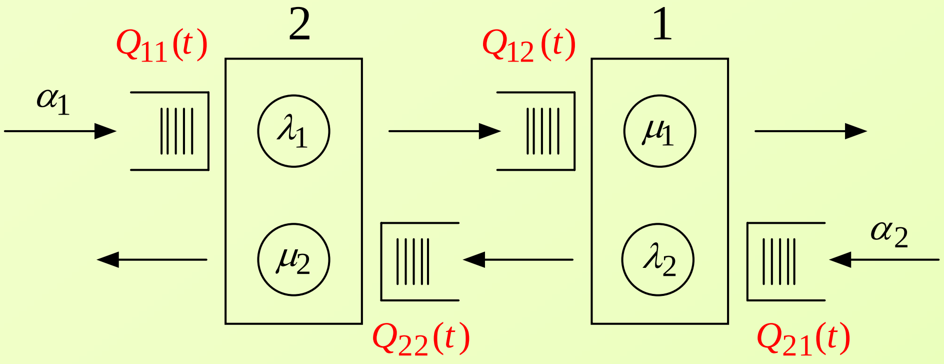
At any time t choose allocation $u(t)$ such that

$$\max Q'(t)Ru(t) \quad \text{s.t.} \quad Au(t) \le \mathbf{1},\ u(t) \ge 0,\ u(t) \text{ is available}$$

In balanced heavy traffic it optimizes the diffusion approximation

Decisions for $\max Q'(t)Ru(t)$

Rybko-Stolyar



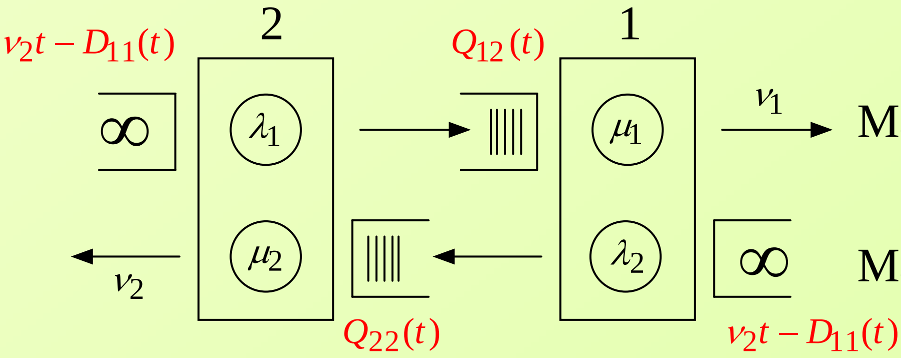
Machine 1 Push if $Q_{11}(t) - Q_{12}(t) < Q_{22}(t)$

Pull if $Q_{11}(t) - Q_{12}(t) > Q_{22}(t)$

Machine 2 Push if $Q_{21}(t) - Q_{22}(t) < Q_{12}(t)$

Pull if $Q_{21}(t) - Q_{22}(t) > Q_{12}(t)$

Infinite supply push pull



Machine 1 Push if $\nu_1 t - D_{11}(t) - Q_{12}(t) < Q_{22}(t)$

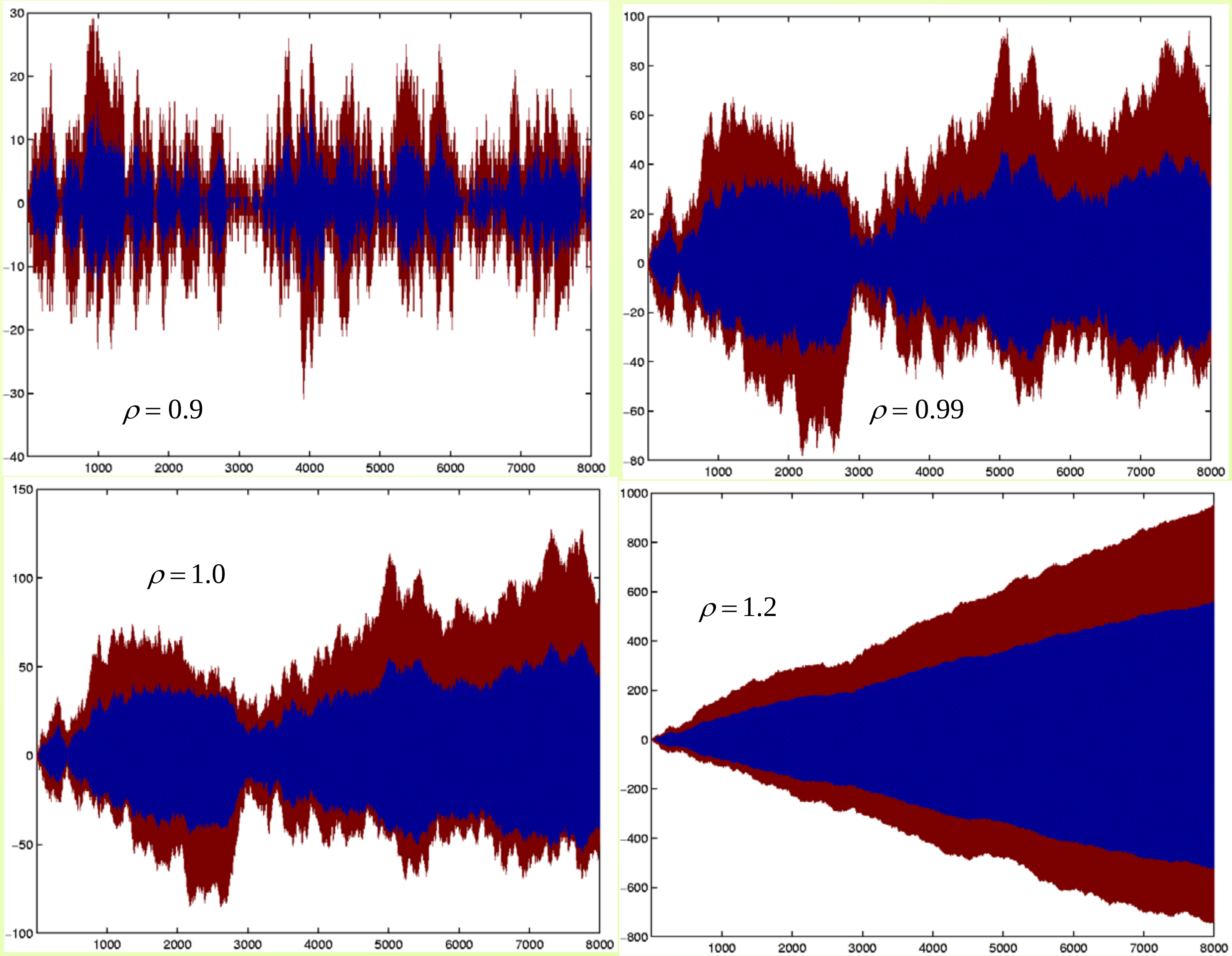
Pull if $\nu_1 t - D_{11}(t) - Q_{12}(t) > Q_{22}(t)$

Machine 2 Push if $\nu_2 t - D_{21}(t) - Q_{22}(t) < Q_{12}(t)$

Pull if $\nu_2 t - D_{21}(t) - Q_{22}(t) > Q_{12}(t)$

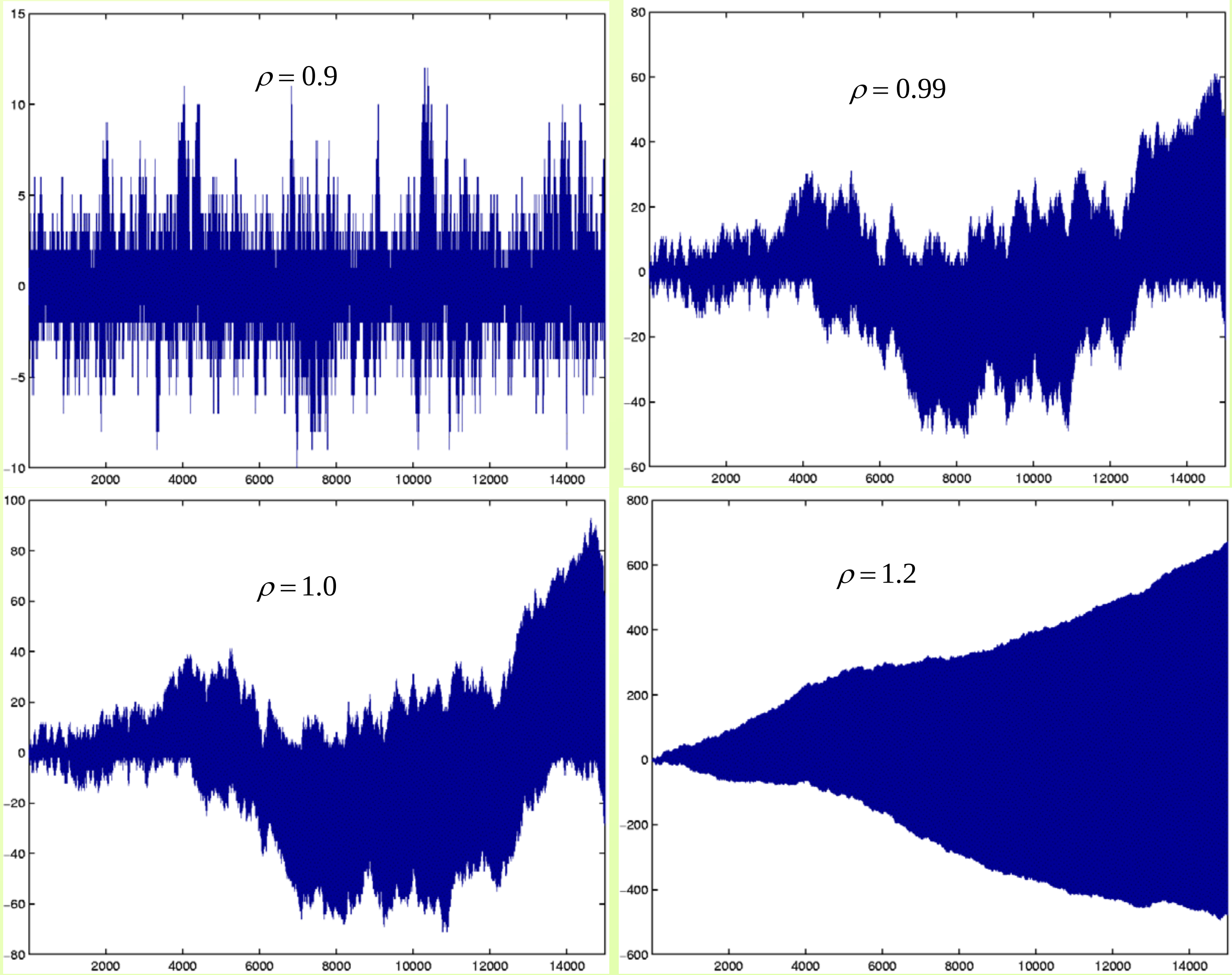
Rybko Stolyar network under max pressure

$$\frac{\lambda}{\mu} = 1.25$$

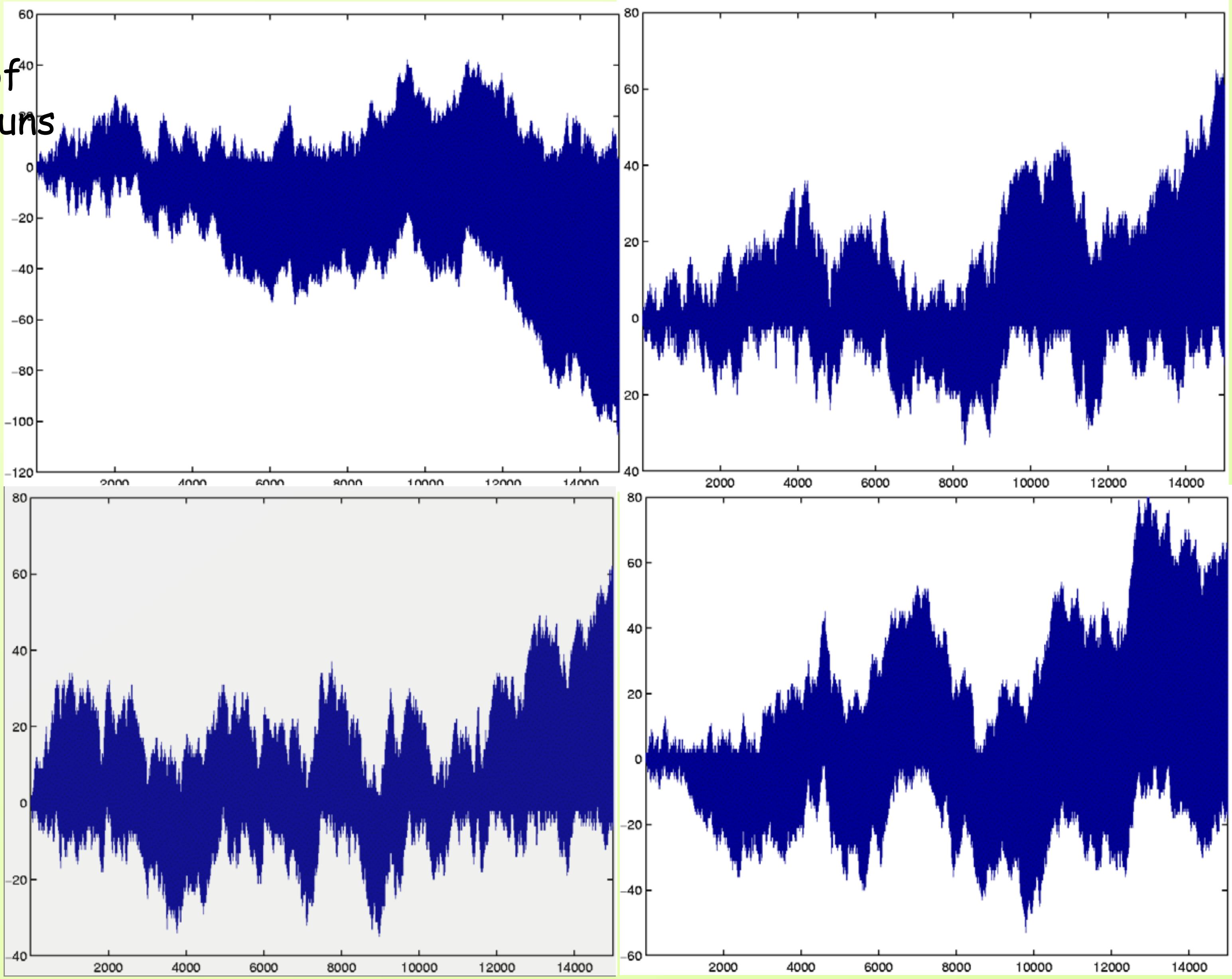


Push Pull network under max pressure

$$\frac{\lambda}{\mu} = 1.25$$



Push Pull under max pressure - trying for full utilization

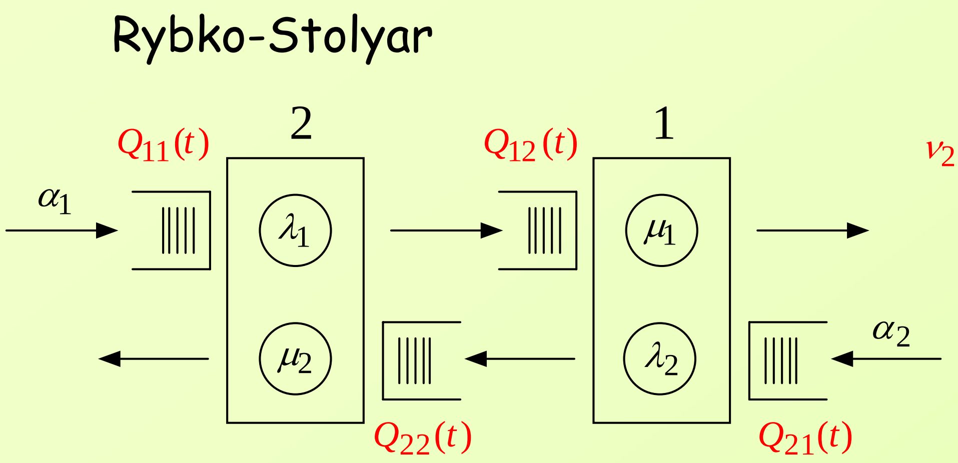


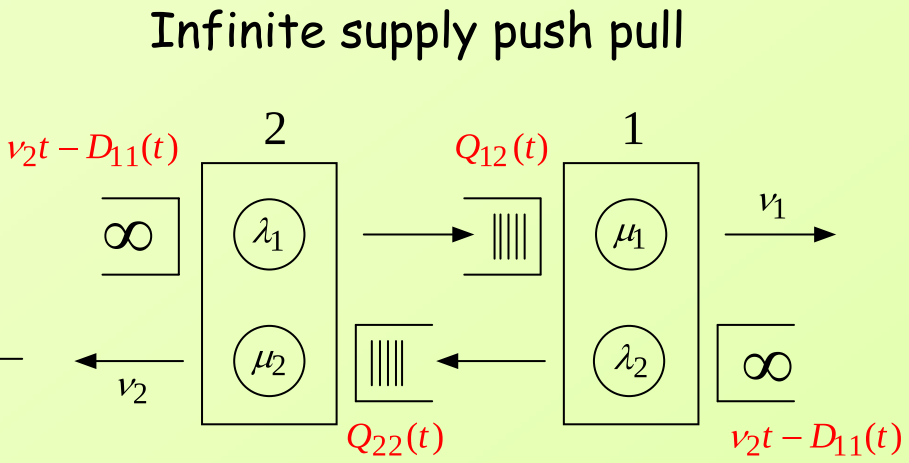
Sample of
4 more runs

$$\frac{\lambda}{\mu} = 1.25$$

$$\rho = 1.0$$

How good is max pressure?





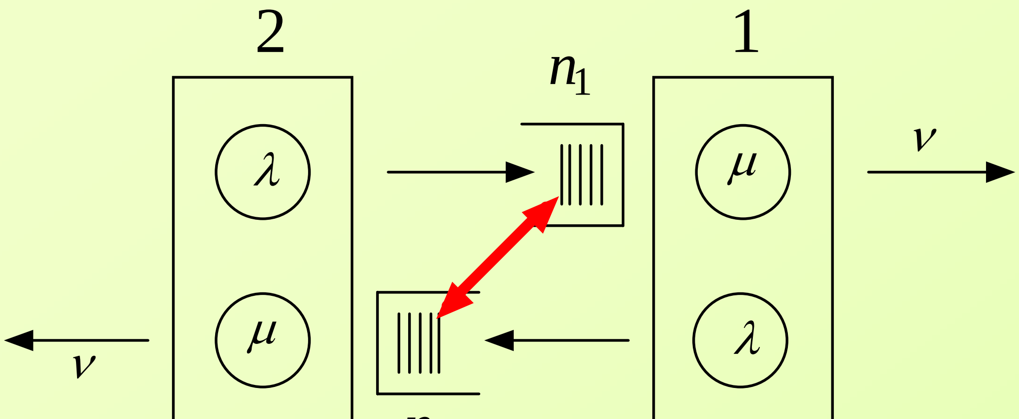
Max pressure attempts to balance the queues, minimize sum of squares

For infinite supply it will balance fluctuations in supply with queues

Using the fixed threshold, fluctuations in supply may be twice as big,
But: Average number in the two queues (with full utilization)

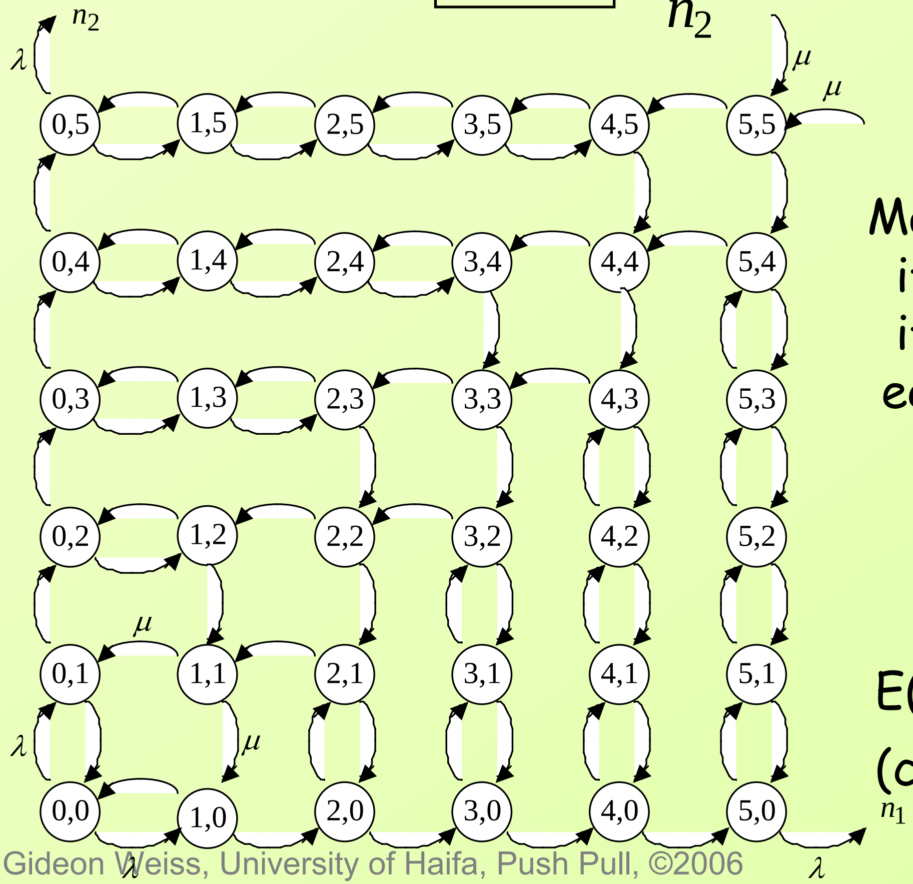
E(queues)= 11.86 $(s_1=s_2=4)$

Push pull system - balancing diagonal policy



Symmetric

$\lambda > \mu$



Machine i : Compare queue to machine j,
if other queue longer - Pull,
if your queue longer - Push,
equal queues pull

$$\frac{\lambda}{\mu} = 1.25$$

E(queues)= 12.37

(compared to 11.86 For fixed threshold)

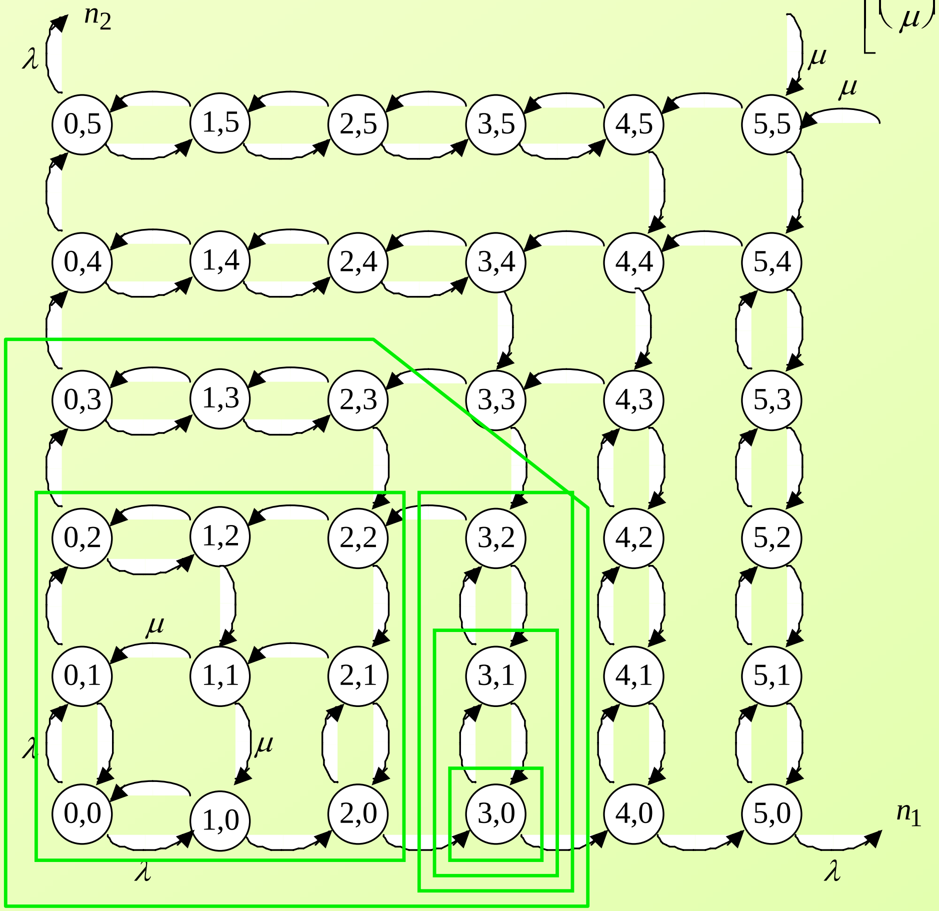
Push pull balancing diagonal policy - steady state

Symmetric

$\lambda > \mu$

$m > n$

$$P_{m,n} = P_{0,0}\left(\frac{\lambda}{\mu}\right)^2 \frac{\left[\frac{\lambda}{\mu}+\frac{\lambda}{\lambda-\mu}\left(\left(\frac{\lambda}{\mu}\right)^{m-1}-1\right)\right]^{m-2}}{\left[\left(\frac{\lambda}{\mu}\right)^{m-1}+\frac{\lambda}{\lambda-\mu}\left(\left(\frac{\lambda}{\mu}\right)^{m-1}-1\right)\right]^{m-1}}\left[2\left(\frac{\lambda}{\mu}\right)^{n}+\frac{\lambda}{\lambda-\mu}\left(\left(\frac{\lambda}{\mu}\right)^{m-2}-1\right)\right]$$

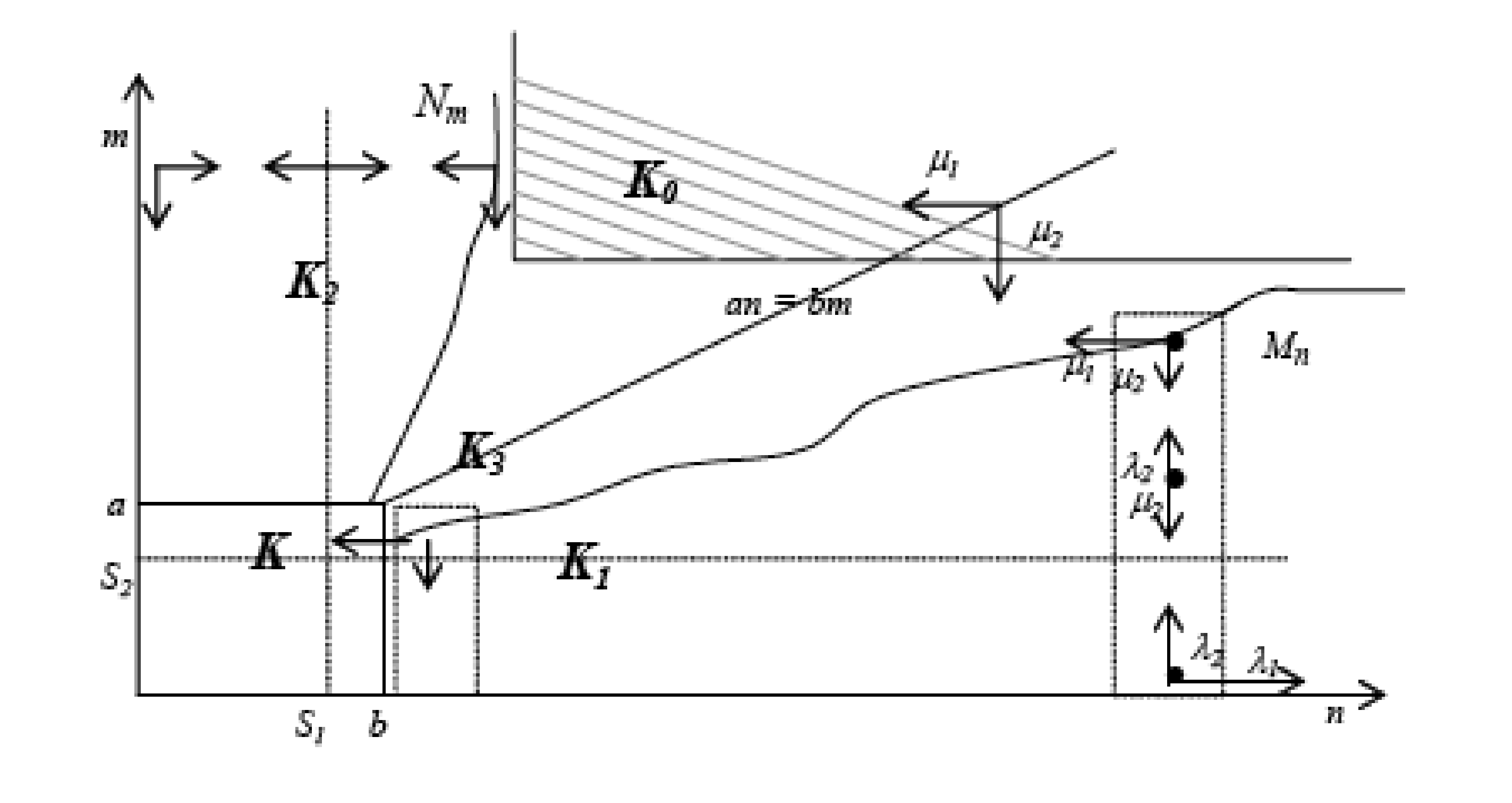


$$P_{m,0} = P_{0,0}\frac{\lambda}{\mu}\left[\frac{\frac{\lambda}{\mu}+\frac{\lambda}{\lambda-\mu}\left(\left(\frac{\lambda}{\mu}\right)^{m-1}-1\right)}{\left(\frac{\lambda}{\mu}\right)^{m-1}+\frac{\lambda}{\lambda-\mu}\left(\left(\frac{\lambda}{\mu}\right)^{m-1}-1\right)}\right]^{m-1}$$

$$P_{m,m} = P_{0,0}\left(\frac{\lambda}{\mu}\right)^2\left[\frac{\frac{\lambda}{\mu}+\frac{\lambda}{\lambda-\mu}\left(\left(\frac{\lambda}{\mu}\right)^{m-1}-1\right)}{\left(\frac{\lambda}{\mu}\right)^{m-1}+\frac{\lambda}{\lambda-\mu}\left(\left(\frac{\lambda}{\mu}\right)^{m-1}-1\right)}\right]^{m-1}$$

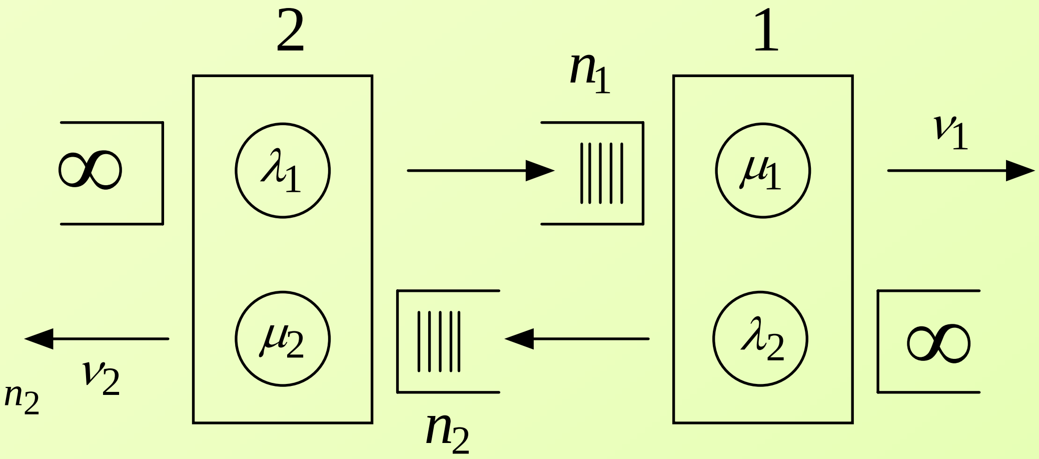
Generalized threshold policies

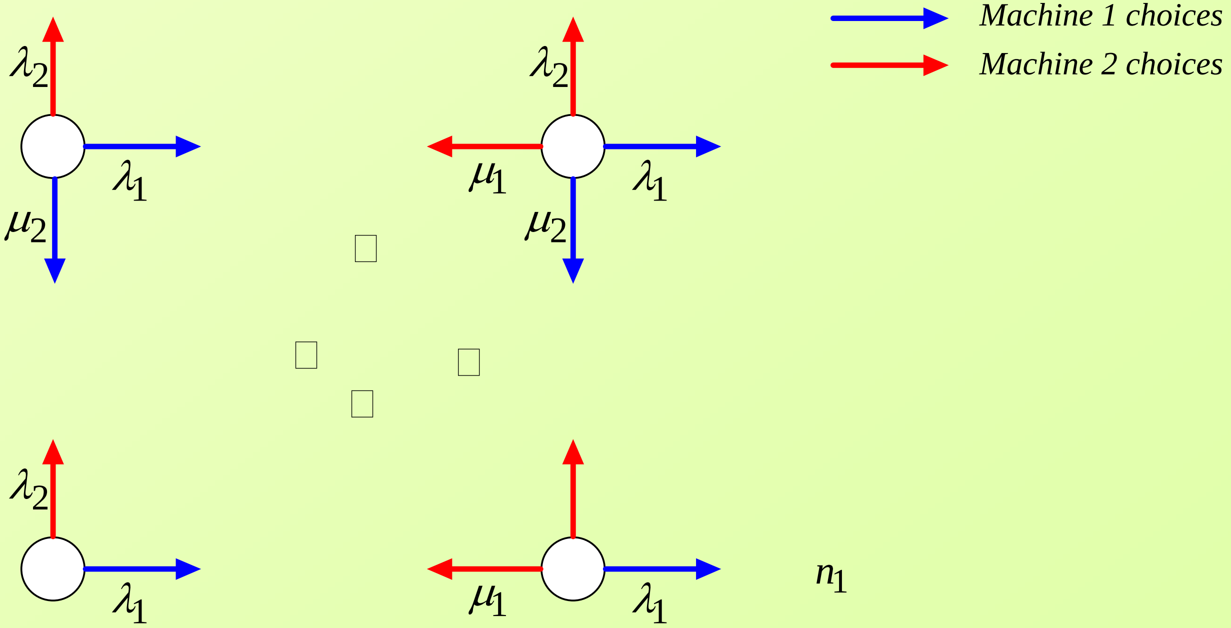
We define two monotone threshold curves, above the levels s_1, s_2 and use those for our policy
We define and appropriate Lyapunov function, and use Foster Lyapunov criterion to show that all of these are stable.



A dynamic programming problem

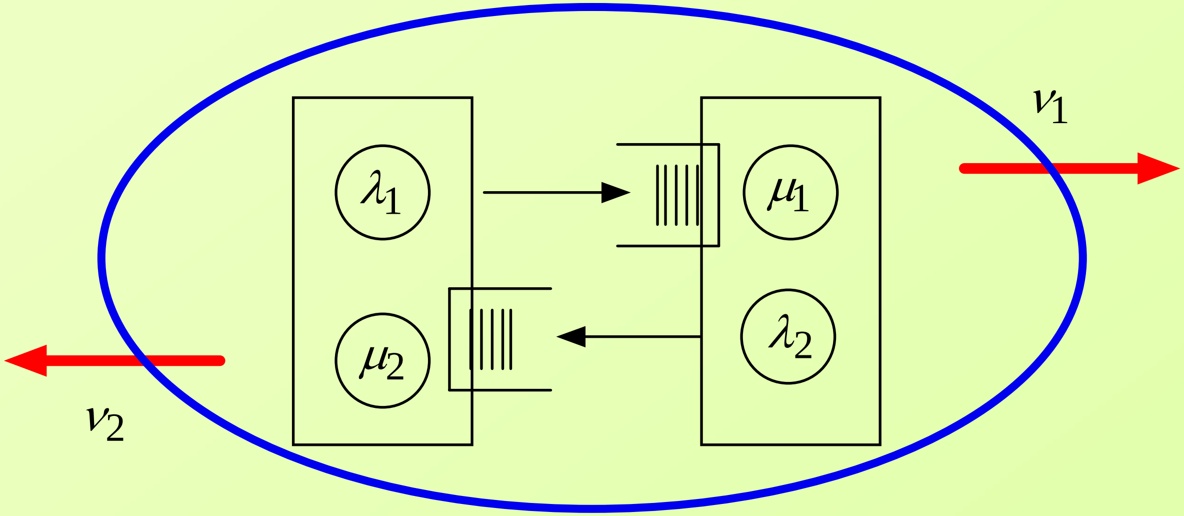
For full utilization, or for a given throughput, find actions to minimize expected queue lengths: Restless bandit indexes?





What next?

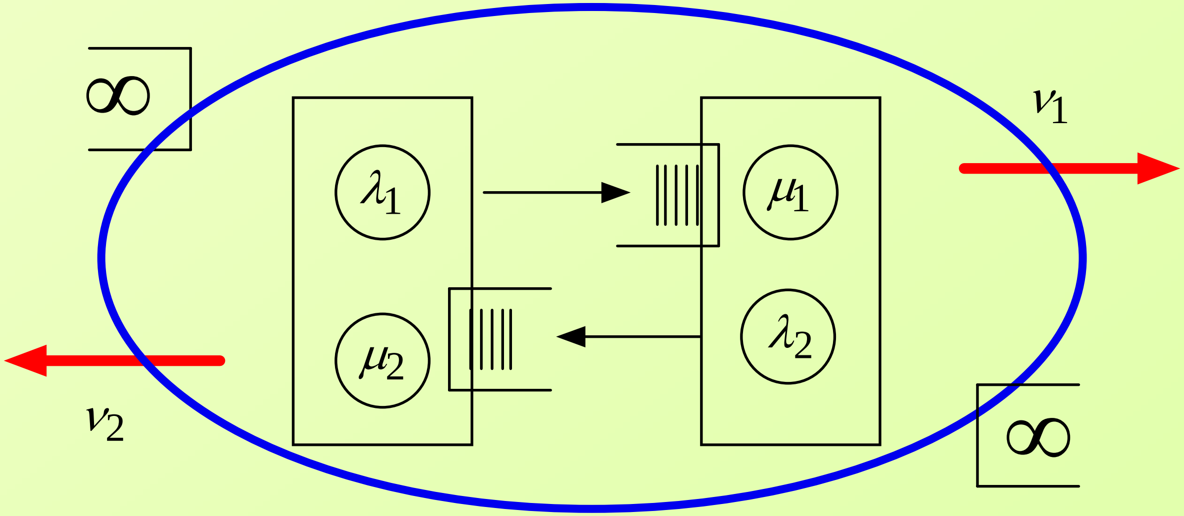
We have looked at a small system that can work at full utilization and not be congested



What next?

We have looked at a small system that can work at full utilization and not be congested

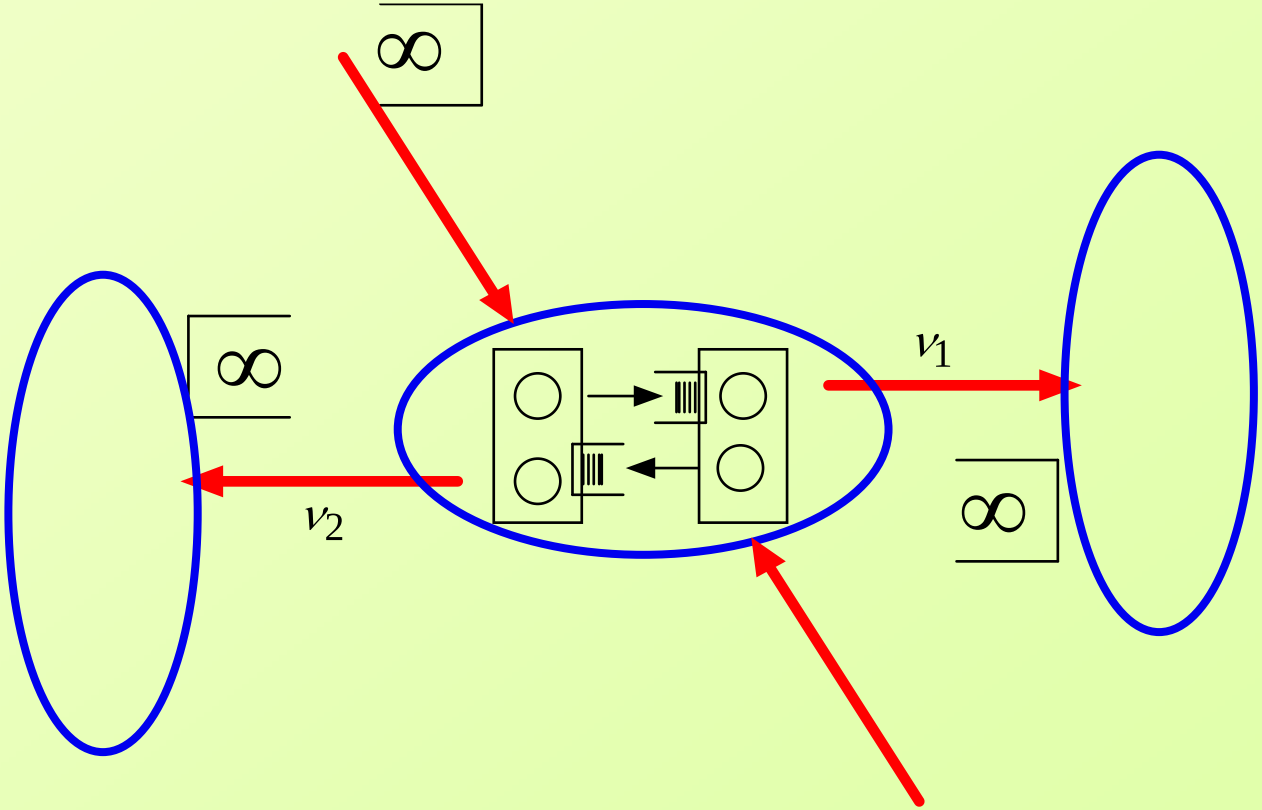
Provided we feed it infinite supply of work



What next?

The challenge is to have a large network working at full utilization with no congestion, on a dynamic basis - by online control which assures supply of work at selected points.

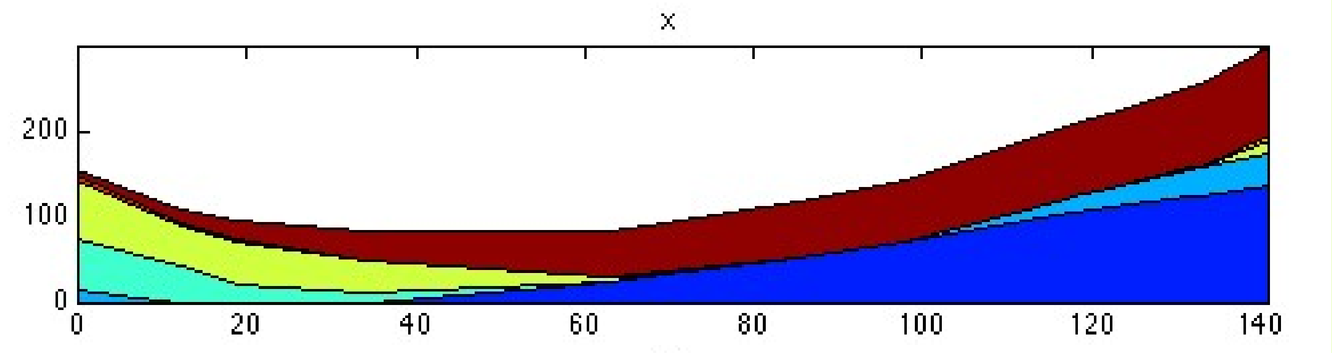
Leads to fluid optimal fluid control, solved as a continuous linear program



What next?

The challenge is to have a large network working at full utilization with no congestion, on a dynamic basis - by online control which assures supply of work at selected points.

Leads to fluid optimal fluid control, solved as a continuous linear program



Summary

We presented a small MCQN with 2 infinite virtual buffers and compared it to the similar Rybko-Stolyar network.

The infinite supply of work, infinite virtual buffers, present a new paradigm for MCQN in balanced heavy traffic

Maximum pressure policies can be adapted to MCQN w infinite virtual buffers and they achieve pathwise stability under full utilization, similar to the Rybko -Stolyar network. However, at full utilization the system will become congested with null recurrent queues that scale as $\sqrt{n}$ to a diffusion

The greater controllability of MCQN with virtual infinite buffers allows full utilization of the system in which all the random fluctuations are pushed to the input and output of the system, and all the internal queues are not congested