Load balancing of elastic data streams in cellular networks

Kaiyuan Wu

Supervisor: Prof. Jorma Virtamo
Instructor: Ph.D. Aalto Samuli

Contents

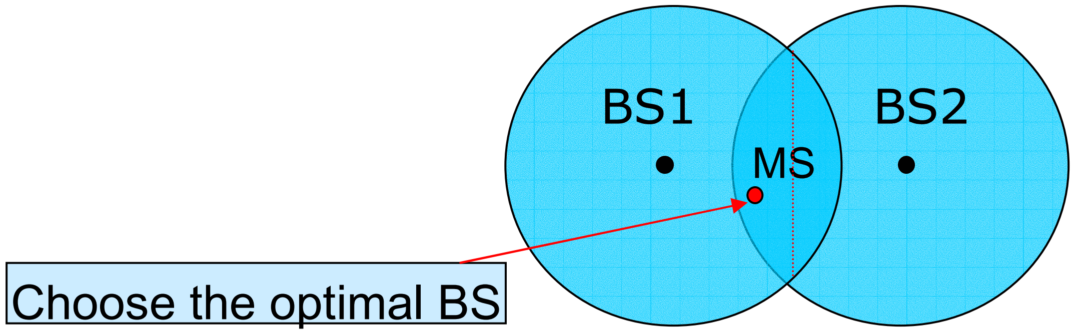
- Background
- Objective
- Model
- Schemes
- Results
- Conclusions

Background

- New data services in cellular networks are boosting like emails, downloading digital documents…
- Each service request corresponds to one elastic flow which probably experiences rate fluctuation
- Load balancing is used to “equalize” the workload according to some specific criteria

BS: base station

MS: mobile station



Objective

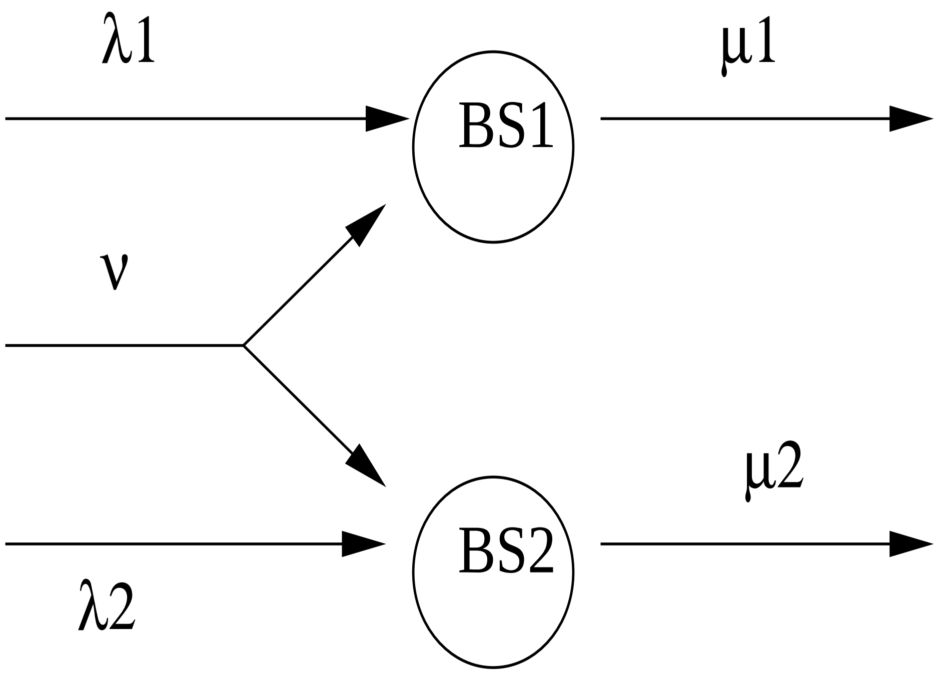
- Propose a load balancing scheme for the elastic flows in the overlapping area between two adjacent cells to minimize the mean flow delay in the packet-switched cellular networks.

Model

- Assumptions
  -new flows arrive according to Poisson processes with rates $\lambda_1, \lambda_2, \nu$, where $\lambda 1 \geq \lambda_2$.

  -the flow size is exponentially distributed with mean $1/\mu_n$.

- Notation:
  - i_n means the flow number in BS_n .



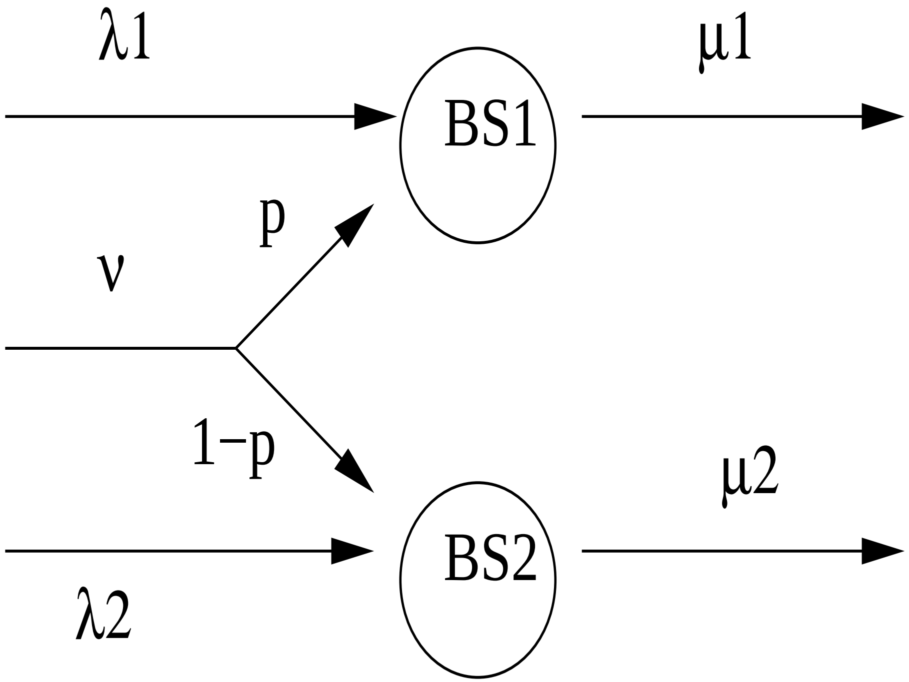
Static routing scheme

- Randomized Routing
  - without state-dependent information
  - new generated flows are routed to

  BS_1 with probability p

  BS_2 with probability $1-p$

  Thus, the system is modeled as two separate $M/M/1$ queues with arriving rate $\lambda_1 + p\nu$ and $\lambda_2 + (1-p)\nu$.



Static policy: ORR

- Optimized Random Routing: minimize the delay in static manner. Basis of later iterations.
- Whole delay is the weighted sum of delay in each subsystem as

$$E[D] = \sum_{i=1}^{2} \frac{\lambda_i + p_i\nu}{\lambda_1 + \lambda_2 + \nu} \cdot \frac{1}{\mu - \lambda_i - p_i\nu}$$

- Optimal probability

$$p* = \begin{cases} \frac{1}{2} - \frac{\lambda_1 - \lambda_2}{2\nu}, & \nu \geq \lambda_1 - \lambda_2 \\ 0, & \nu < \lambda_1 - \lambda_2 \end{cases}$$

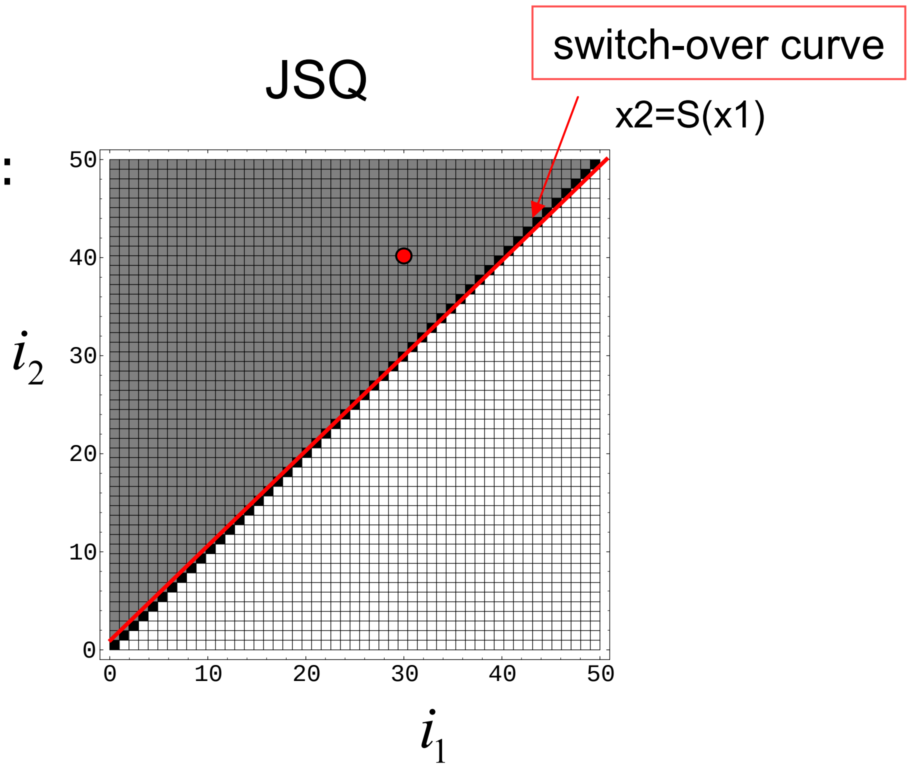
Load Balance case (LB)

No Load Balance case (NLB)

ORR balances the load as evenly as possible

Some dynamic schemes

- When a flexible flow enters, we can apply:
- Join the shortest queue (JSQ): literally explicit. Just pick the base station associated with less flow number
- Least ratio routing (LRR): select the base station with the less relative load, i_n / μ_n . If the capabilities of the two stations are of the same, it evolves to JSQ case.



Decision:
dark grey: to BS1
light grey: to BS2

Policy iteration

- In brief, the optimal policy is derived via policy iteration (developed in Markov Decision Process)

  -fix the immediate cost rate R_i and average collecting time τ_i for each state i

  -choose an basic policy α as a starting-point

  -solve the relative value v_i for each state along with the average cost rate $\overline{R}$ from Howard equations.

  - In state $\hat{i}$, select a better station associated with less expected value $R \cdot \tau_{\hat{i}}(\alpha) - \overline{R} \cdot \tau_{\hat{i}}(\alpha) + v_{\hat{j}}(\alpha)$, where $\hat{i} = (i_{1,} i_2)$.

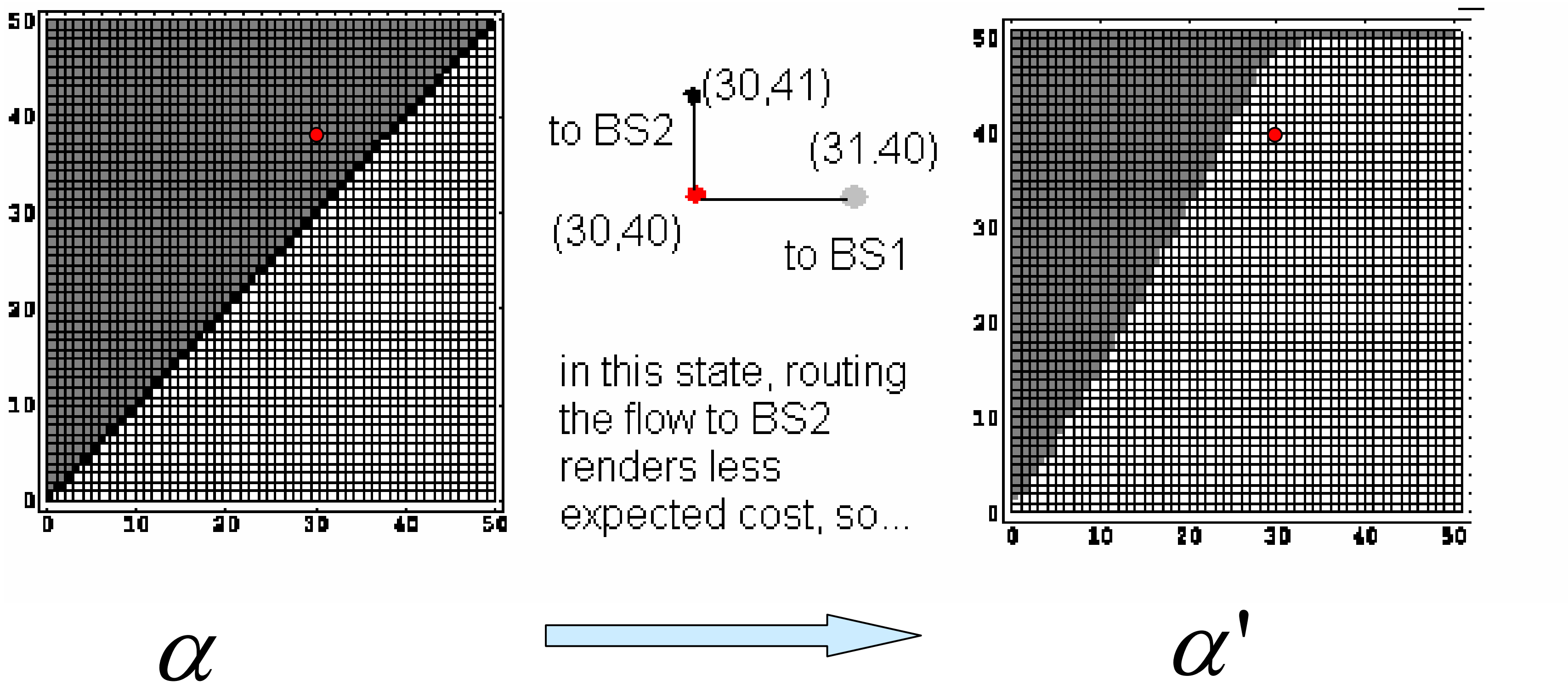
  -a better policy α' (with less cost) is derived if we apply the selection in each state

  -an optimal policy α^* appears until cost cannot be further optimized

Policy iteration (cont.)

A typical example of policy iteration.

The expected cost: $R \cdot \tau_{\hat{i}}(\alpha) - \overline{R} \cdot \tau_{\hat{i}}(\alpha) + v_{\hat{j}}(\alpha)$



FPI & FPI*

- First policy iteration (FPI) is based on the ORR and it is derived by the policy iteration
- Iterated policy α' is given as,

$$\alpha'(i_1, i_2) = \begin{cases} 1, & v_1(i_i+1) - v_1(i_i) \le v_2(i_2+1) - v_2(i_2) \\ 2, & v_1(i_i+1) - v_1(i_i) \succ v_2(i_2+1) - v_2(i_2) \end{cases}$$

Marginal cost

- FPI is given as, $t(i_1, i_2) = \dfrac{i_1+1}{\mu - \lambda_1 - p_1\nu} - \dfrac{i_2+1}{\mu - \lambda_2 - p_2\nu}$

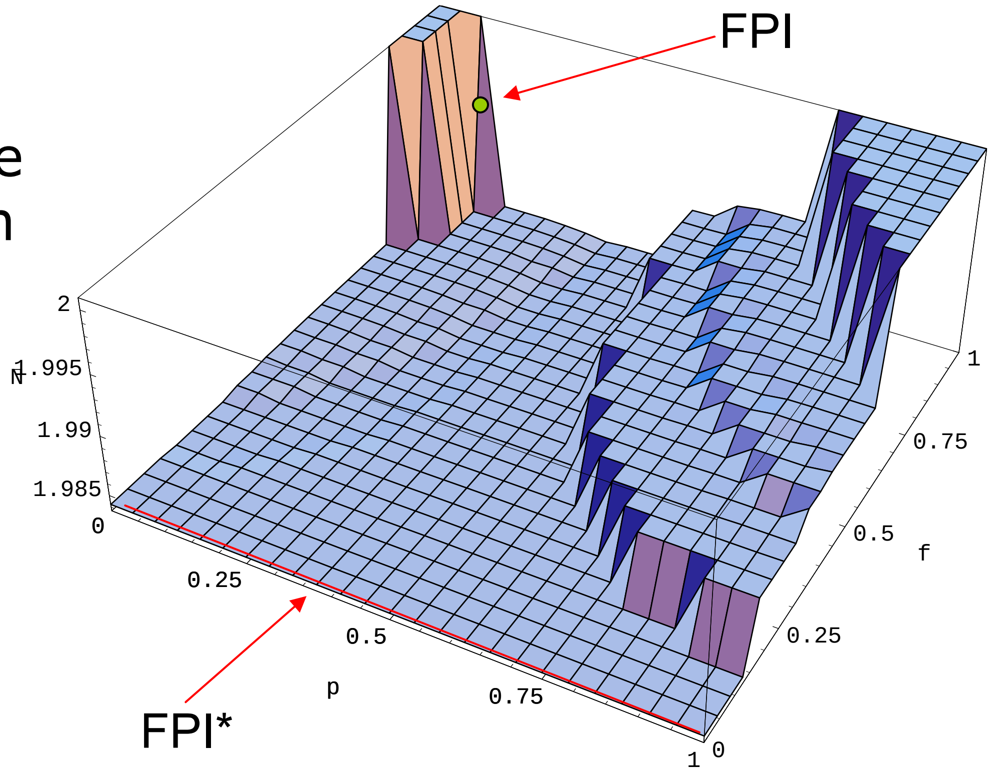
- If we simply ignore the flexible flow, FPI* is

$$t(i_1, i_2) = \frac{i_1+1}{\mu - \lambda_1} - \frac{i_2+1}{\mu - \lambda_2}$$

Marginal cost:
cost of accepting an additional flow

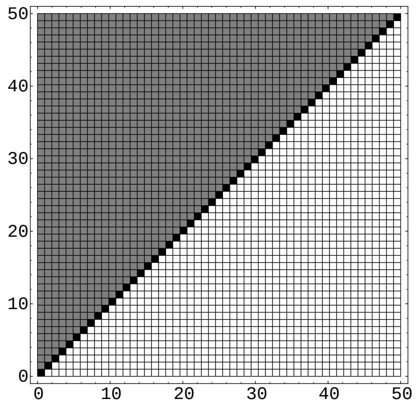
Basic policy optimization

- If we only admit new generated flows with probability f and routes the accepted flows to BS1 with probability p.
- FPI corresponds to basic policy (1,p*)
- FPI* corresponds to basic policy (0,0)
- Each combination of f and p forms an unique basic policy
- However, f=0 is optimal

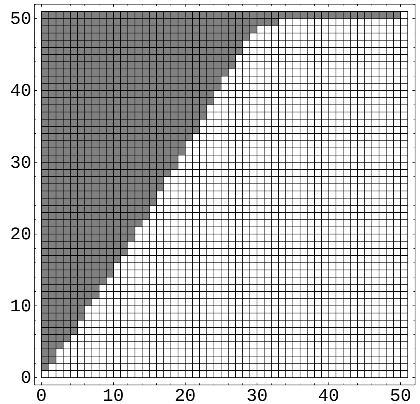


$$\mu 1 = \mu 2 = 20, \lambda 1 = 10, \lambda 2 = 6, \nu = 5$$

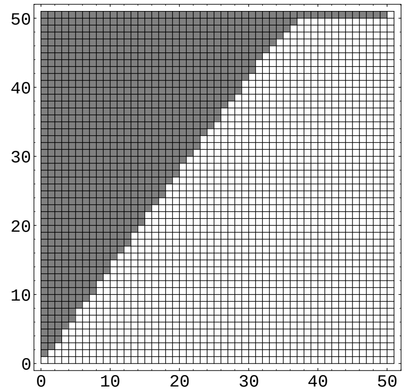
Simulation results



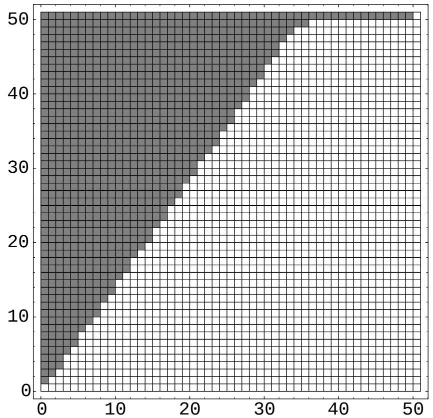
FPI: First Policy Iteration



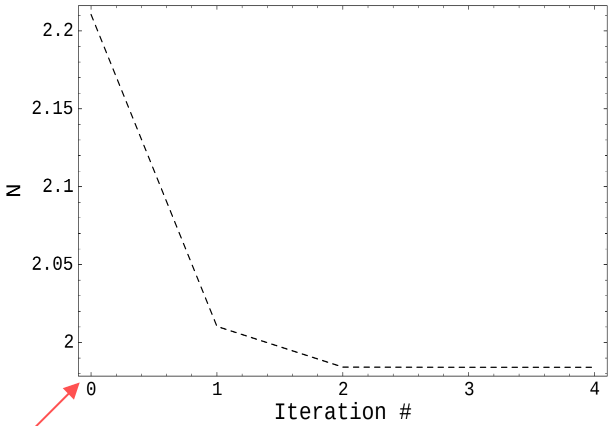
2nd iteration



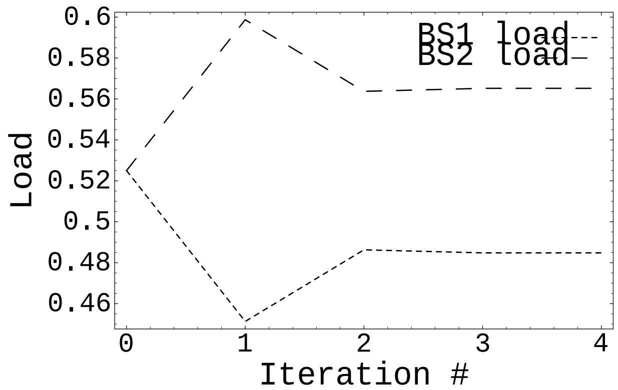
3rd iteration



4th iteration

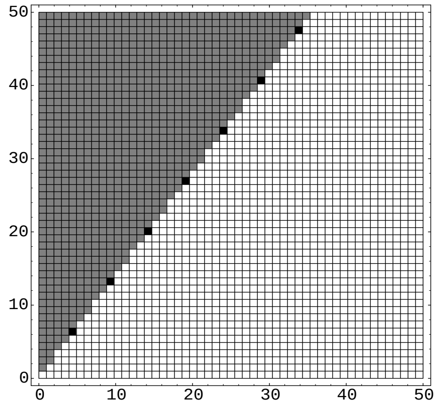


OR
R

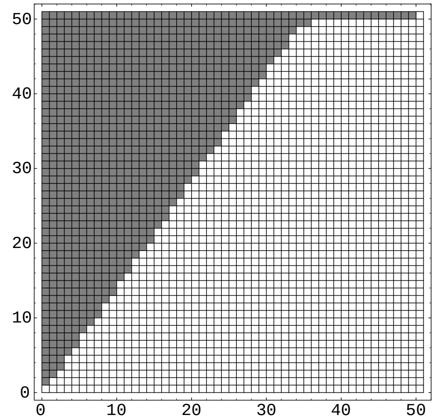


$$\mu 1 = \mu 2 = 20, \lambda 1 = 10, \lambda 2 = 6, \nu = 5$$

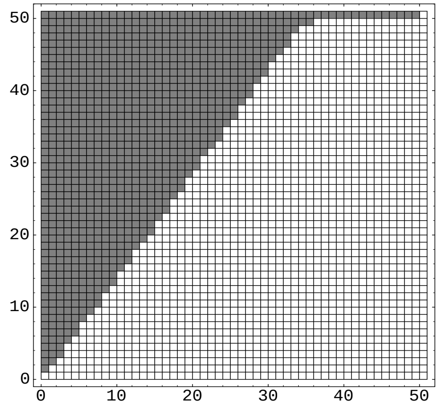
Simulation results (cont.)



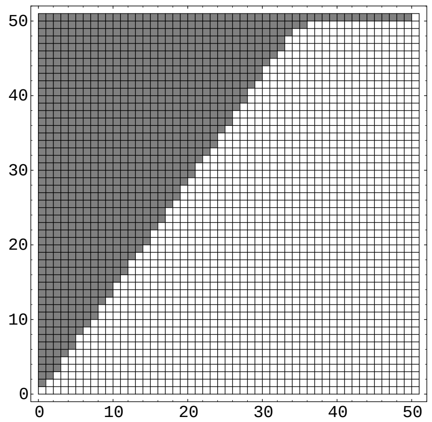
FPI*



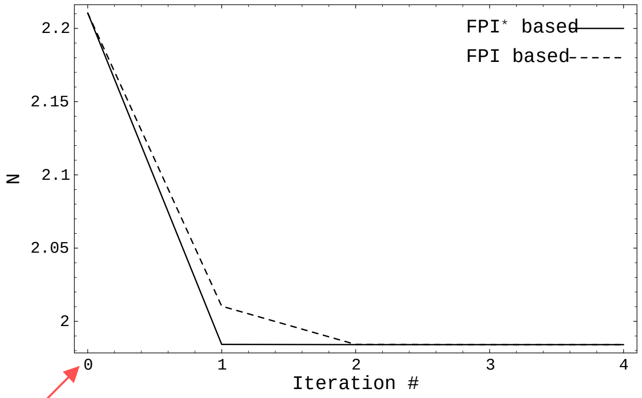
2nd iteration

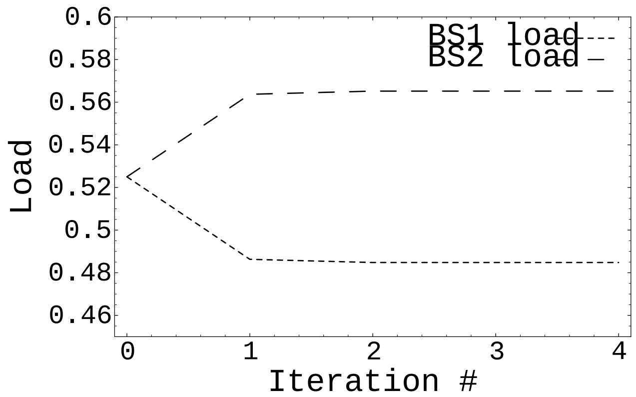


3rd iteration



4th iteration



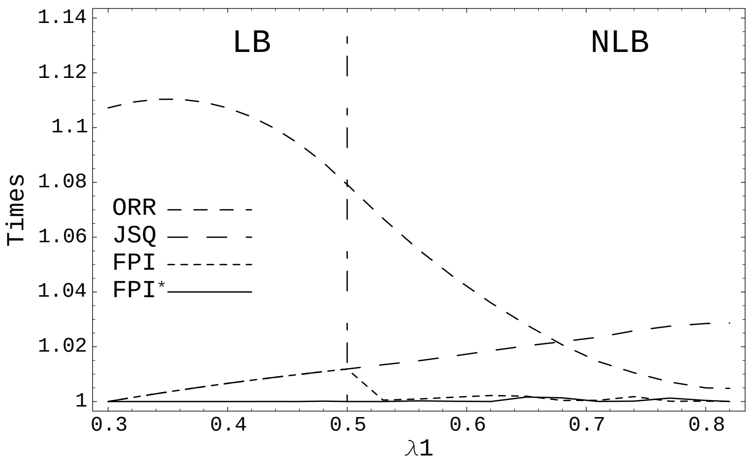


OR
R

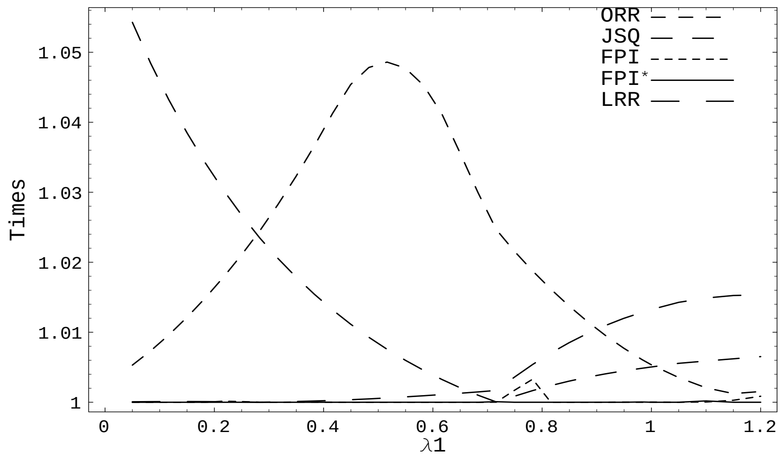
$$\mu 1 = \mu 2 = 20, \lambda 1 = 10, \lambda 2 = 6, \nu = 5$$

Systematic study

- Experiments are carried out to evaluate the performances of different policies. Results are normalized by the optimal policy result.
- In both symmetric and asymmetric case, FPI* closely resembles the optimal policy.



$\mu 1 = \mu 2 = 1.0, \lambda 2 = 0.3, \nu = 0.2, \lambda 1 \in [0.3, 0.9]$



$\mu 1 = 1.5, \mu 2 = 1.0, \lambda 2 = 0.2, \nu = 0.1, \lambda 1 \in [0.05, 1.2]$

Conclusion

- Optimal policy can be characterized by a switch-over curve
- FPI* scheme is proposed as a rather robust scheme
- Flow delay can be alleviated significantly

Thank you!