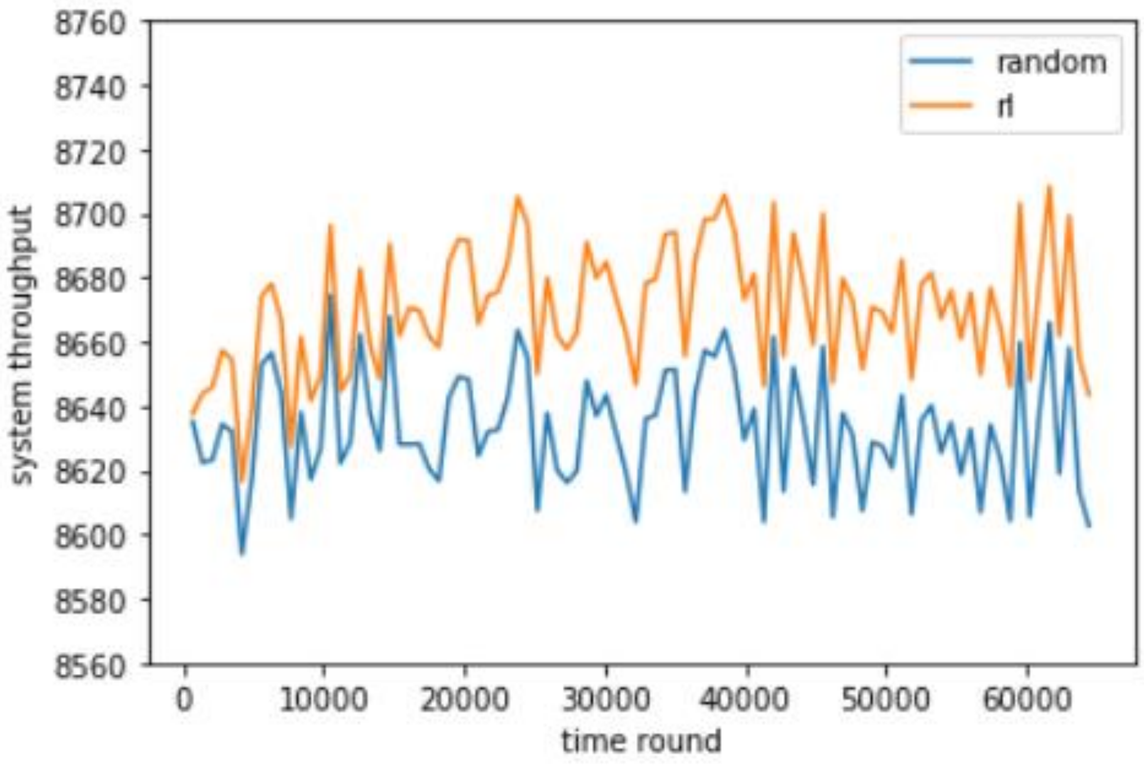
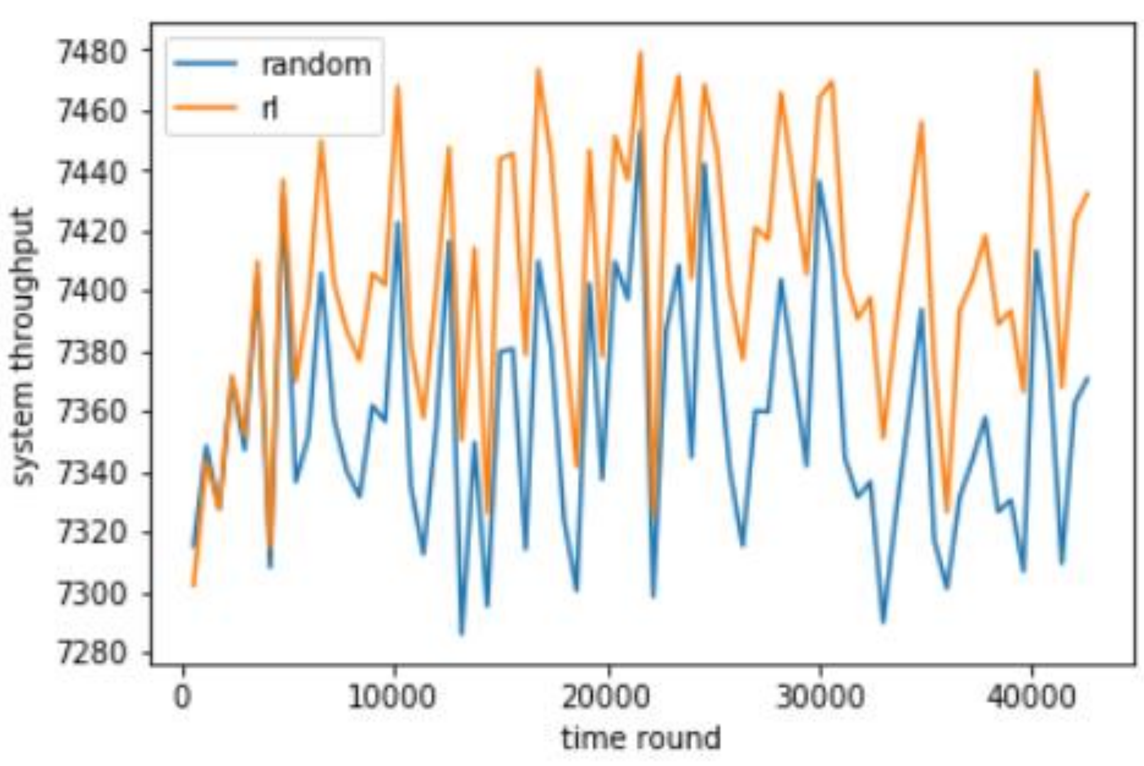
Experiment and Formulation

2018 12.27

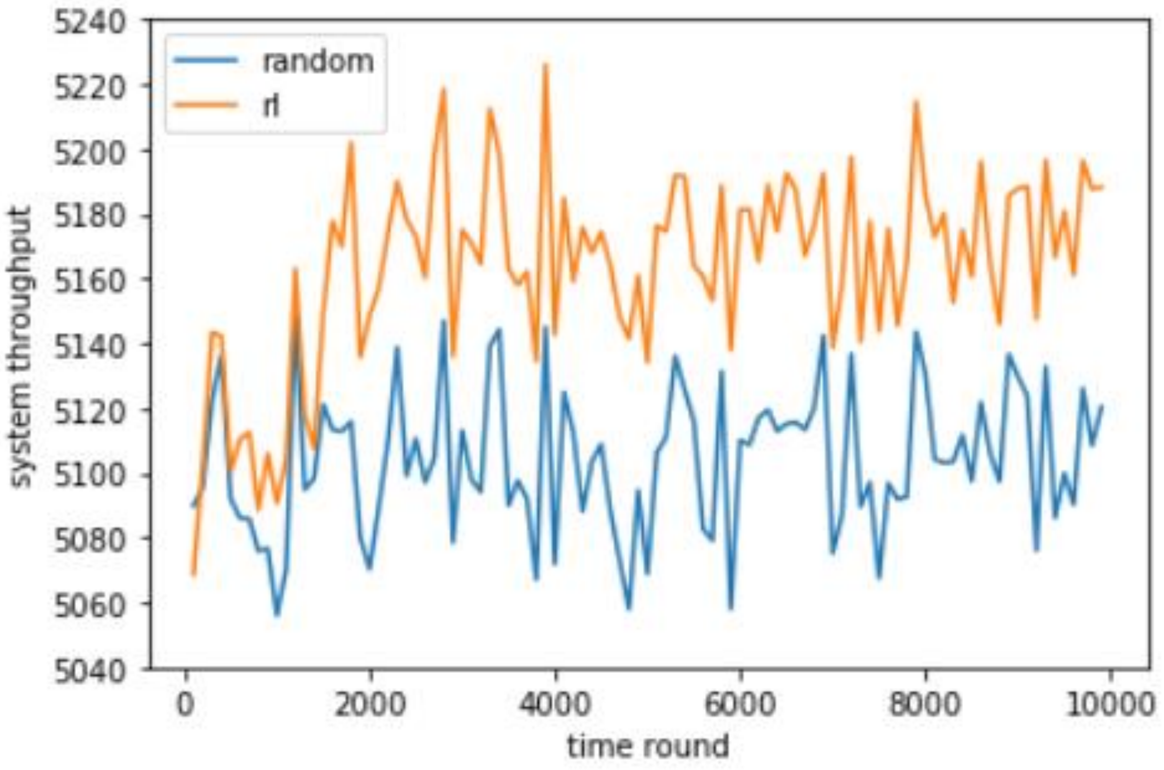
- Topology with 100 nodes, 4 T type nodes (at the top). Set capacity of T-T peer links to be large enough so that they won't be too crowded.

- When inject about 2000 flows into the topology: (dense)

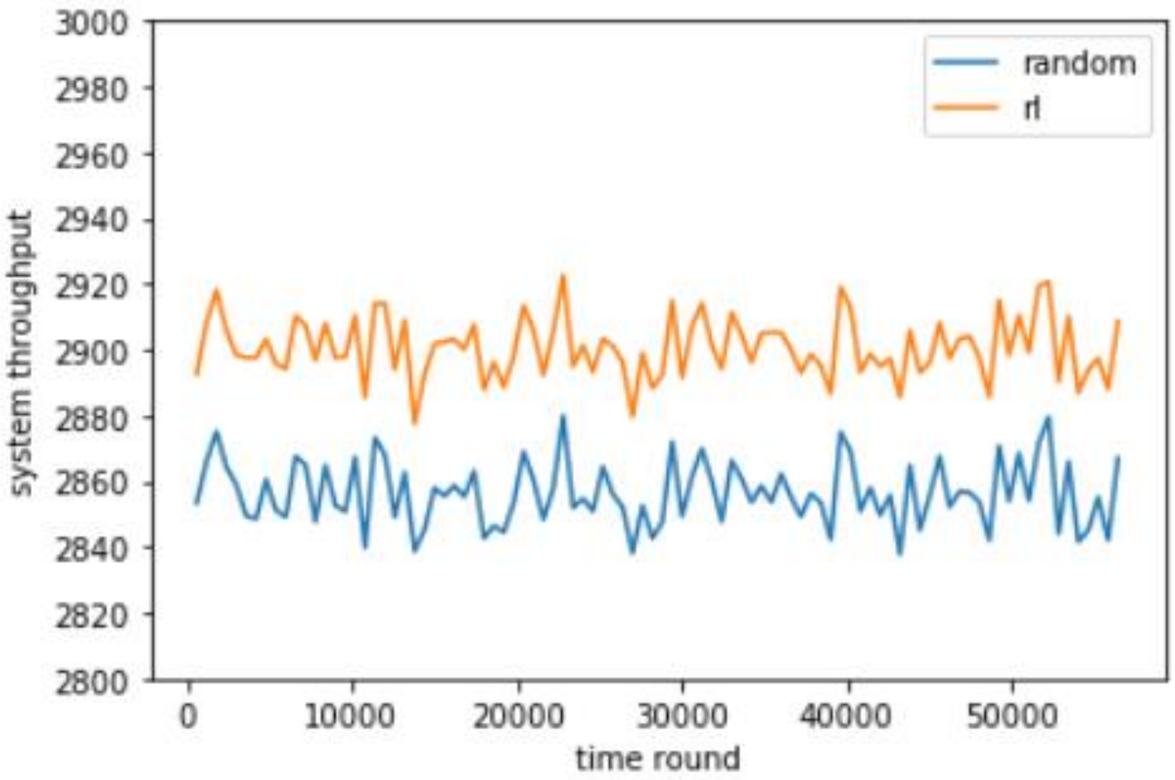


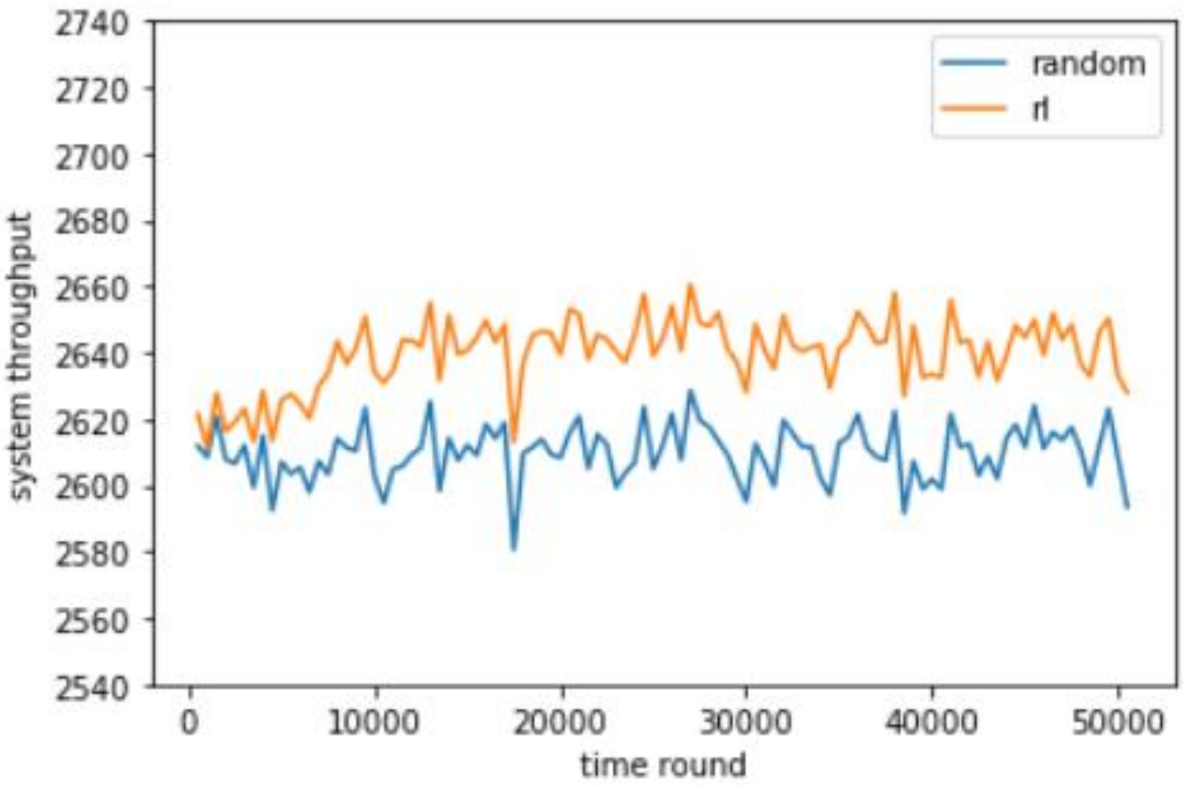


- When inject 1000 flows into the topology:

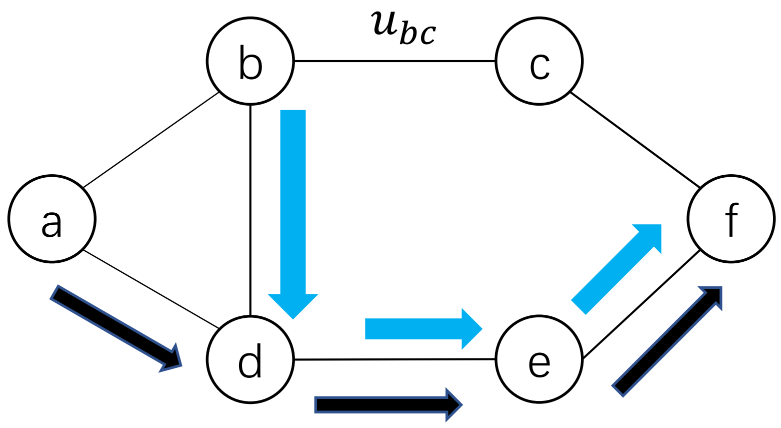


- When inject 400 flows into the topology: （sparse）





Optimal Routing Formulation



- Network topology has flow $f \in F$, each with end-to-end rate v^f, source node s^f and destination node t^f.
  The goal is to maximize optimization variable

$$\max\left(\sum_{f \in F} v^f\right)$$

- With constrains:

$$\sum_{j:(i,j)\in E} x_{ij}^f - \sum_{i:(j,i)\in E} x_{ji}^f = \begin{cases} 0, & for\ all\ i \in V - s^f - t^f \\ -v^f, & for\ i = t^f \\ v^f, & for\ i = s^f \end{cases}$$

$$0 \le \sum_{f \in F} x_{ij}^f \le u_{ij}\ ,\ (i,j) \in E$$

Using Linear Programming

- Change the formulation to relaxation format:

$$\begin{cases} \min c^T x \\ Ax = b \\ x \geq 0 \end{cases}$$

- Change the model and introduce relaxation variables L to it:

$$\begin{cases} \min -\sum_{f\in F,(s^f,i)\in E} x^f_{s^f i}\,, & for\ \forall f \in F,\ (s^f,i) \in E \\ \sum_{j:(i,j)\in E} x^f_{ij} - \sum_{i:(j,i)\in E} x^f_{ji} = 0, & for\ \forall f \in F, i \in V - s^f - t^f \\ \sum_{(i,j)\epsilon E} x^f_{ij} + L^f = u_{ij}\,, & for\ \forall f \in F \\ x^f_{ij} \geq 0, & for\ \forall f \in F, (i,j) \in E \end{cases}$$

- Applying simplex algorithm to it:

1. Find first basic solution based on relaxation formulation. Assume relaxation variables as base variables, others as non-base.

2. Following Bland rule:
   - In the objective function, select the first non-base variable with a negative coefficient to be substituted
   - In the constraint set, select the first base variable that is the tightest constraint to substitute.

   Do pivot operation.

3. Repeat step2 until no non-base variable with a negative coefficient can be found in the objective function.

- Using python library *pulp* to solve.