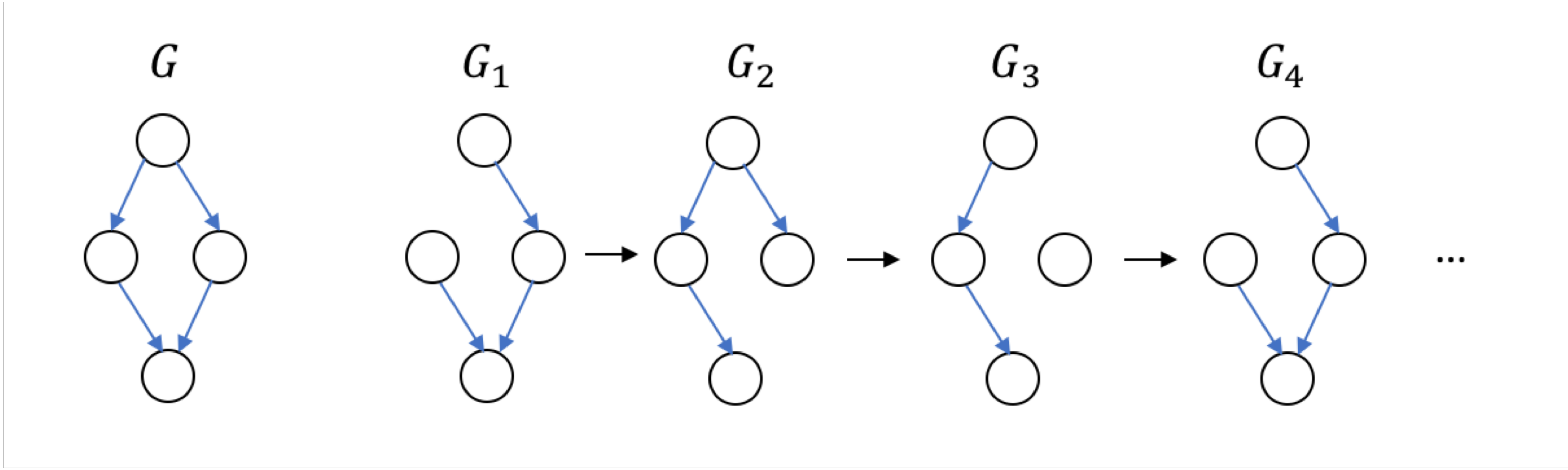
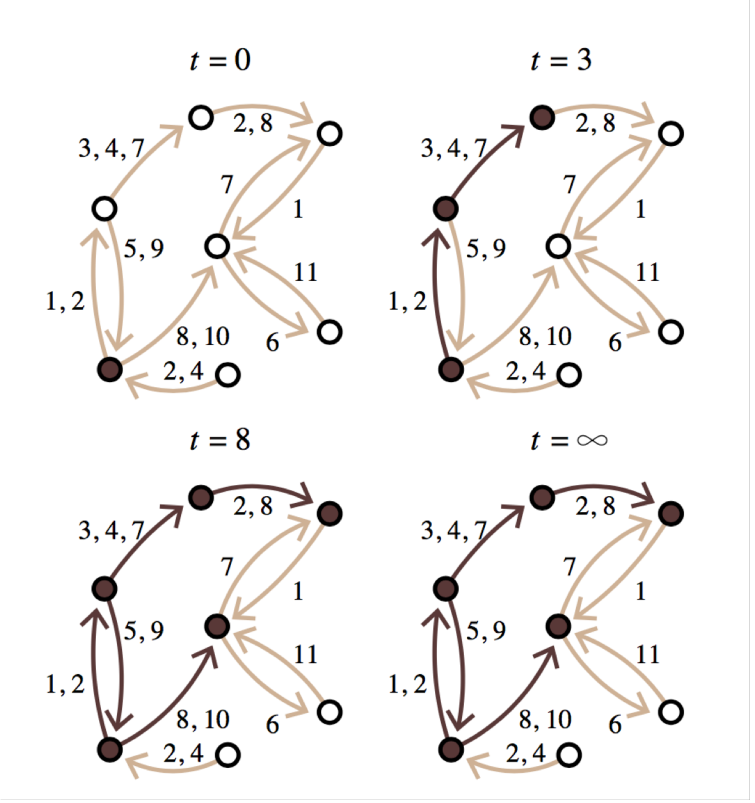


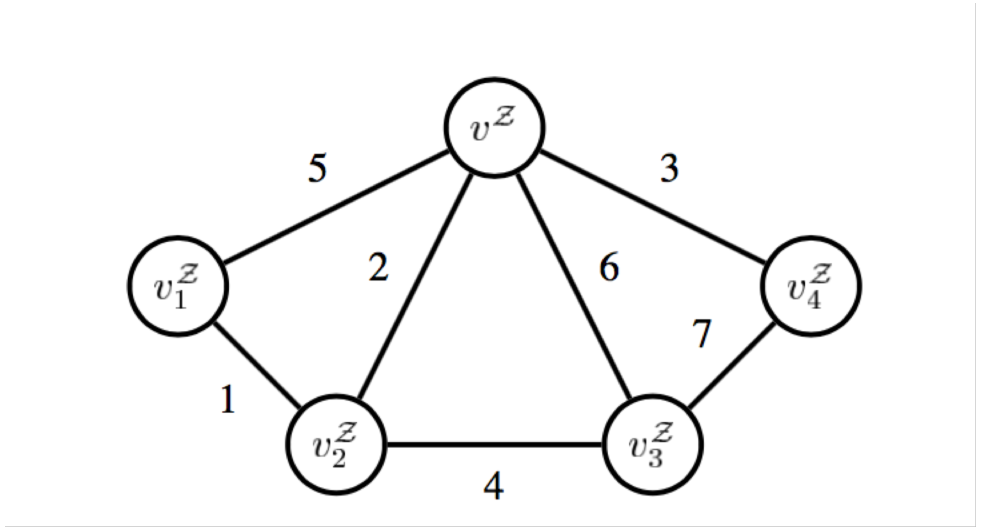
Analysis of Probabilistic Temporal Networks

Xiang Fu, Shangdi Yu

Context: Deterministic Temporal Networks

- Edges are labeled: time at which it's available
  - Certainly appear in labeled timeslots
  - Does not appear in other timeslots

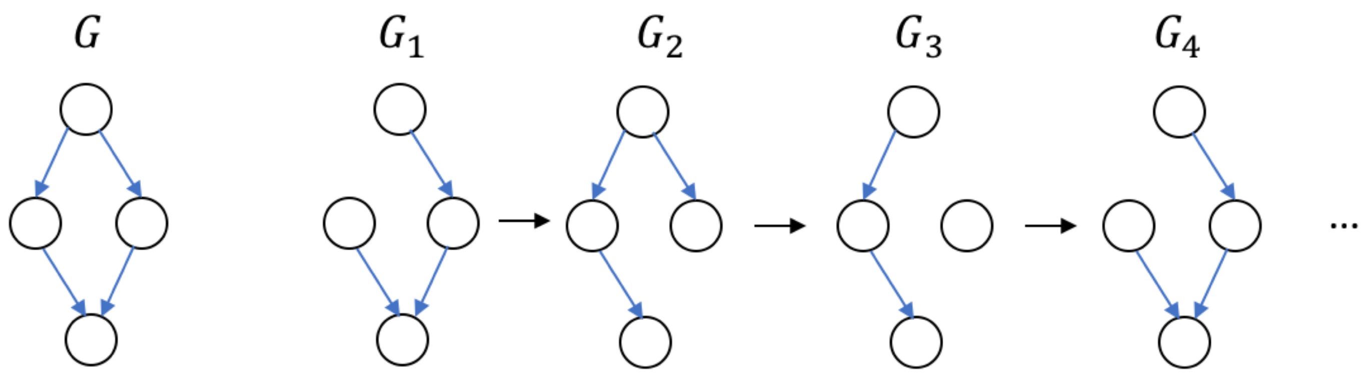




Question: What is the behavior of a non-deterministic temporal networks?

- Encode uncertainty into the network by giving each edge a probability of being "available" during each time slot

$$PTN(G,p): G_1, G_2, ..., G_t, ...,$$
$$G = (V, E), \text{ underlying directed graph}$$
$$p, \text{ probability}$$
$$c_{ij:i\neq j}, \text{ traveling cost; } c_{ii}, \text{ stalling cost}$$
$$G_t = (V_t, E_t)$$



Routing Problem in PTN:
s -> t with smallest total expected cost (traveling & stalling)

- Take Available Shortest Path policy (TASP)

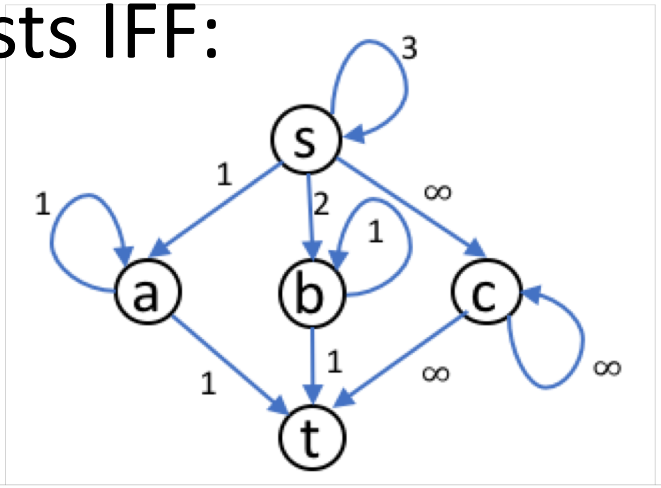
$$v^* = \begin{cases} argmin_{v \in S_{it}} l(v) + c_{iv} & \text{if } \exists v \in E_t \text{ such that } l(v) + c_{iv} \text{ is finite} \\ i & \text{otherwise} \end{cases}$$

- Always Wait Policy (AW)

$$v^* = \begin{cases} v' & \text{if } \exists v' \in E_t \text{ such that } c_{iv} + l(v') = l(i)) \\ i & \text{otherwise} \end{cases}$$

Optimal Routing Policy

- Expected Cost Based Topological Order Exists IFF:
  - Stalling cost <= traveling cost
  - i.e. $c_{kk} \leq c_{ij:i \neq j} \; \forall \, i, j, k \in V$



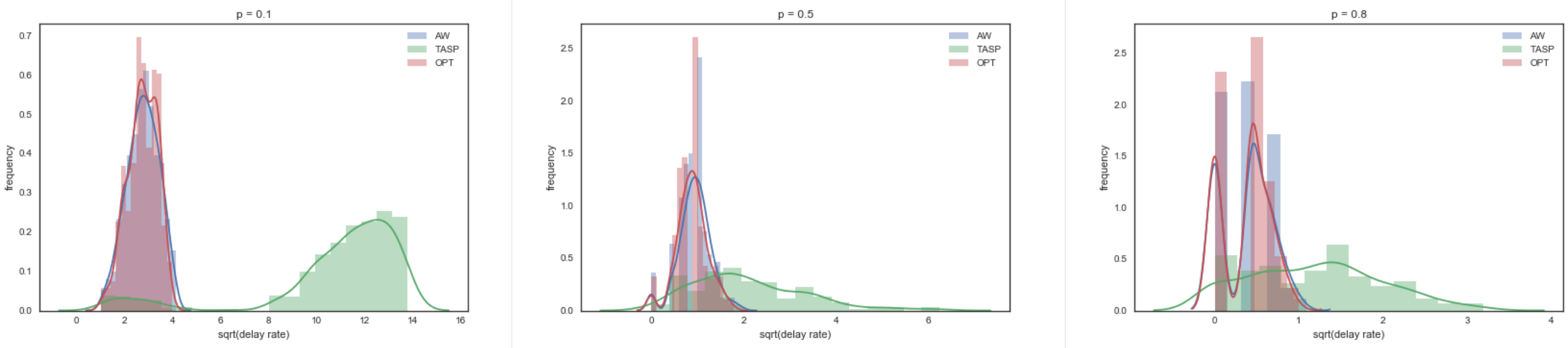
- Use DP to calculate the smallest expected cost w_v from each node v to the destination node t
- Get w_v correct for at least one more node each iteration
- Policy: go to the available node with smallest expected cost + traveling/stalling cost

Algorithm 1 Generating Optimal Routing Table

```
Set w[t] = 0, set w[v] = ∞ for all v ∈ V, v ≠ t
flag = 1
while flag = 1 do
    flag = 0
    for all v ∈ V do
        w'[v] ← (Σ_{x∈N(v), c_vx+w[x]<c_vv+w[v]} (c_vx+w[x])P(x) + (1 − Σ_{x∈N(v), c_vx+w[x]<c_vv+w[v]} P(x)) c_vv) / Σ_{x∈N(v), c_vx+w[x]<c_vv+w[v]} P(x)
        if w'[v] < w[v] then
            flag = 1
            w[v] ← w'[v]
return w
```

Simulation - k-degree random graph

- Stalling cost <= traveling cost instance
- TASP performs significantly worse
- AW is almost as good as OPT

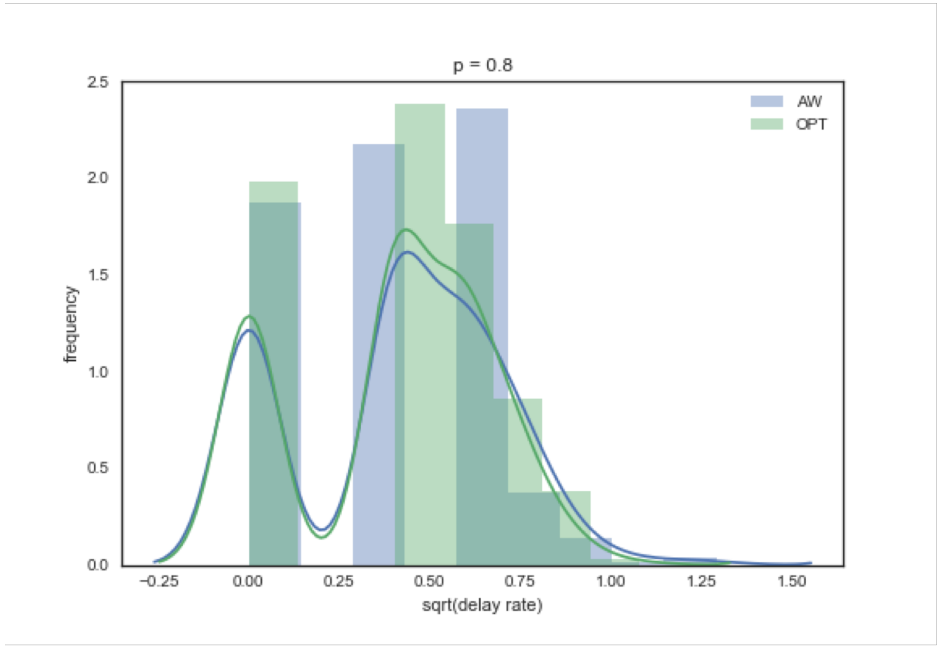


N = 100, k=3, iteration = 500, cost ~1 on k-degree random graph.

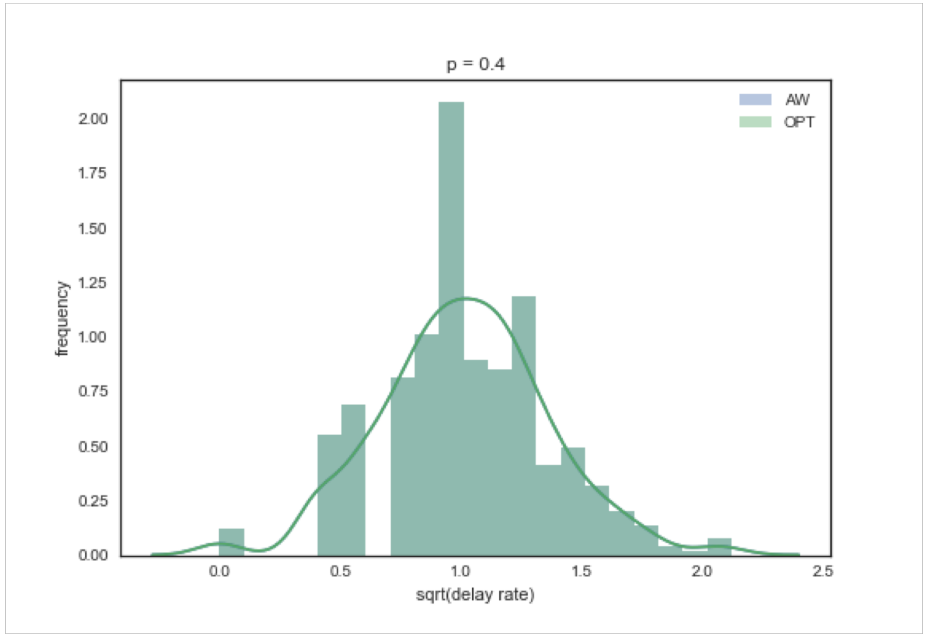
Simulation - k-degree random graph

- Stalling cost <= traveling cost instance

OPT out perform AW instance



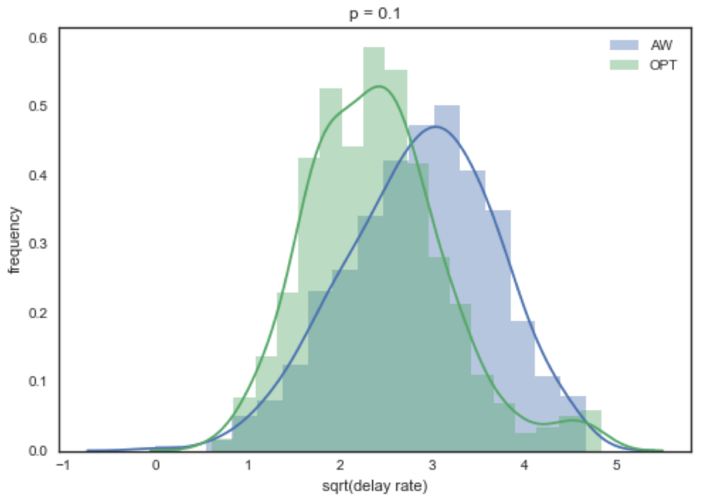
AW makes the same choices as OPT



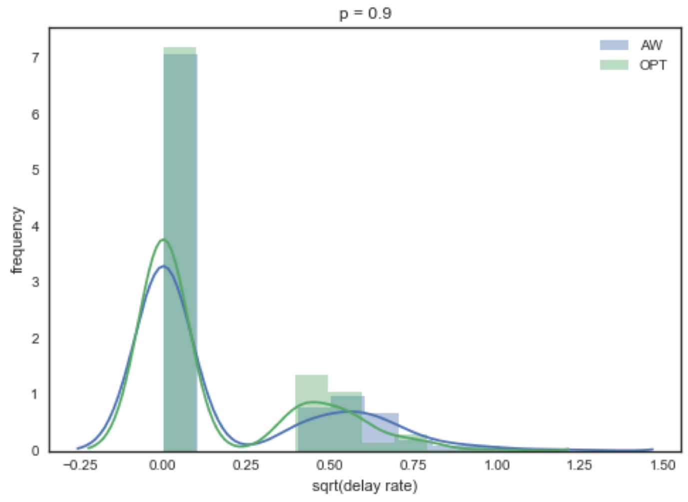
N = 500, k=3, iteration = 500, cost ~1 on k-degree random graph.

Simulation – OPT when assumption violated

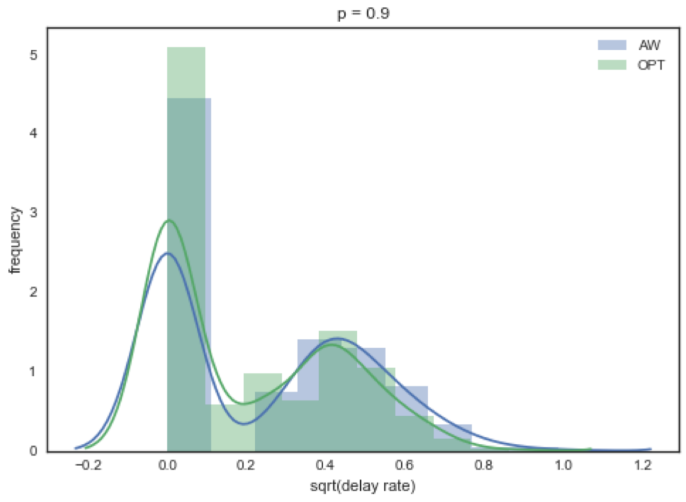
- Stalling cost NOT NECESSARILY <= traveling cost
- As a heuristic, ”OPT” out performs AW
- Due to small-world property in k-deg random graph, the number of nodes does not have a significant influence on the distribution of the sqrt(delay rate)



N = 30, p =0.1



N = 30, p =0.9

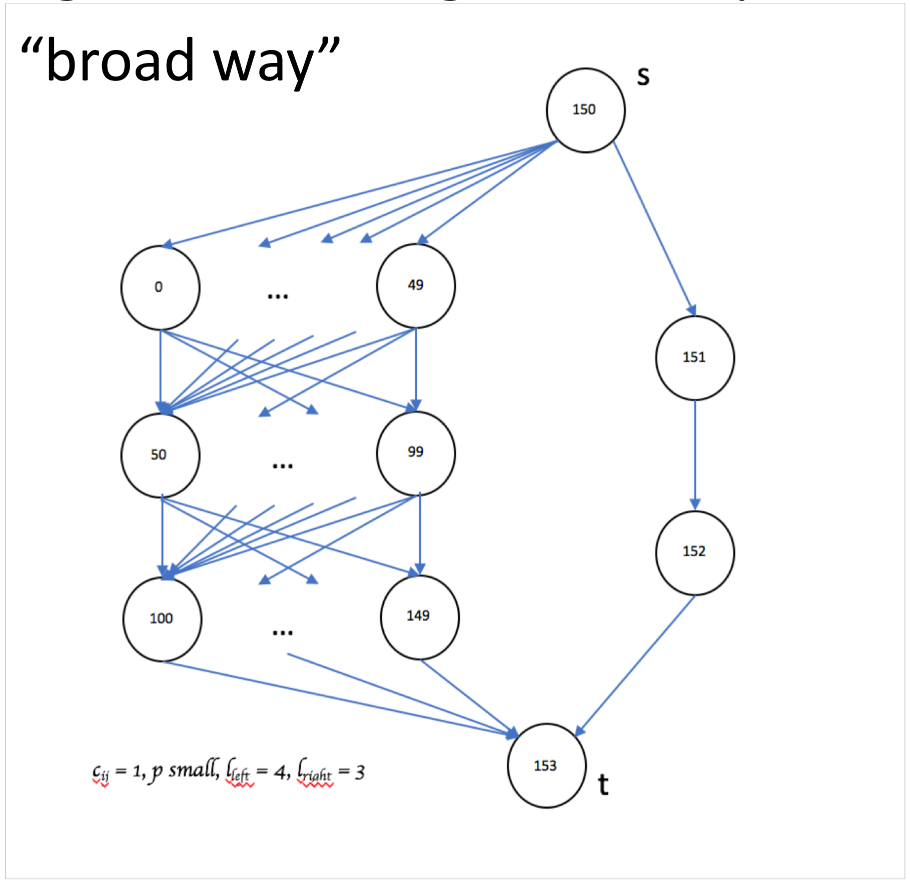


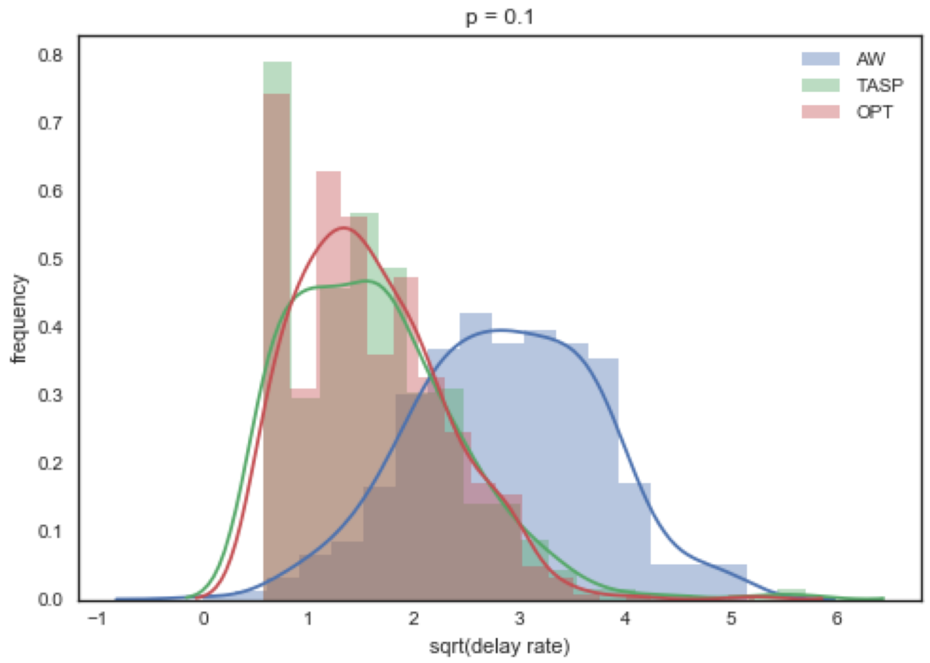
N = 500, p =0.9

k=3, iteration = 500, cost [1,5] on k-degree random graph.

Simulation – graph where AW Policy is bad

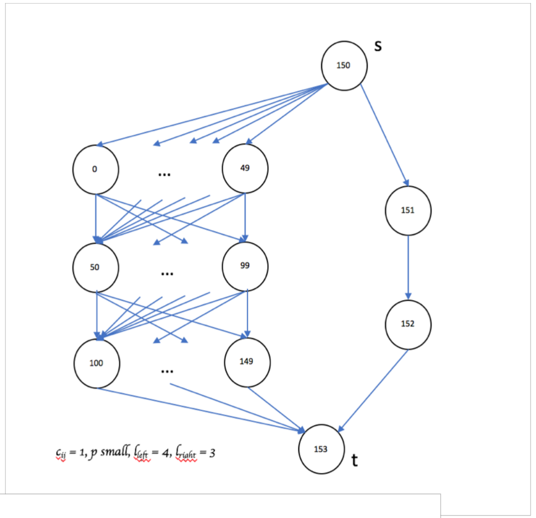
- AW Policy is good in general, but bad in this particular instance:
  - All edge costs/stalling costs = 1, p is small
  - The “broad way”

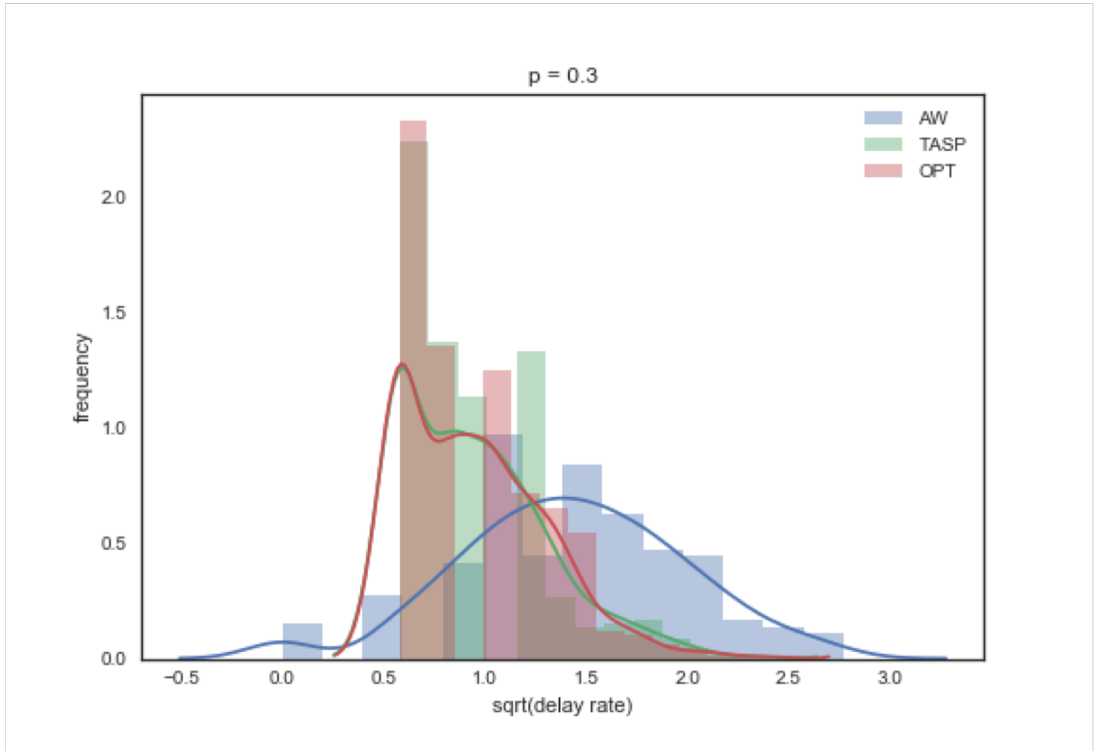


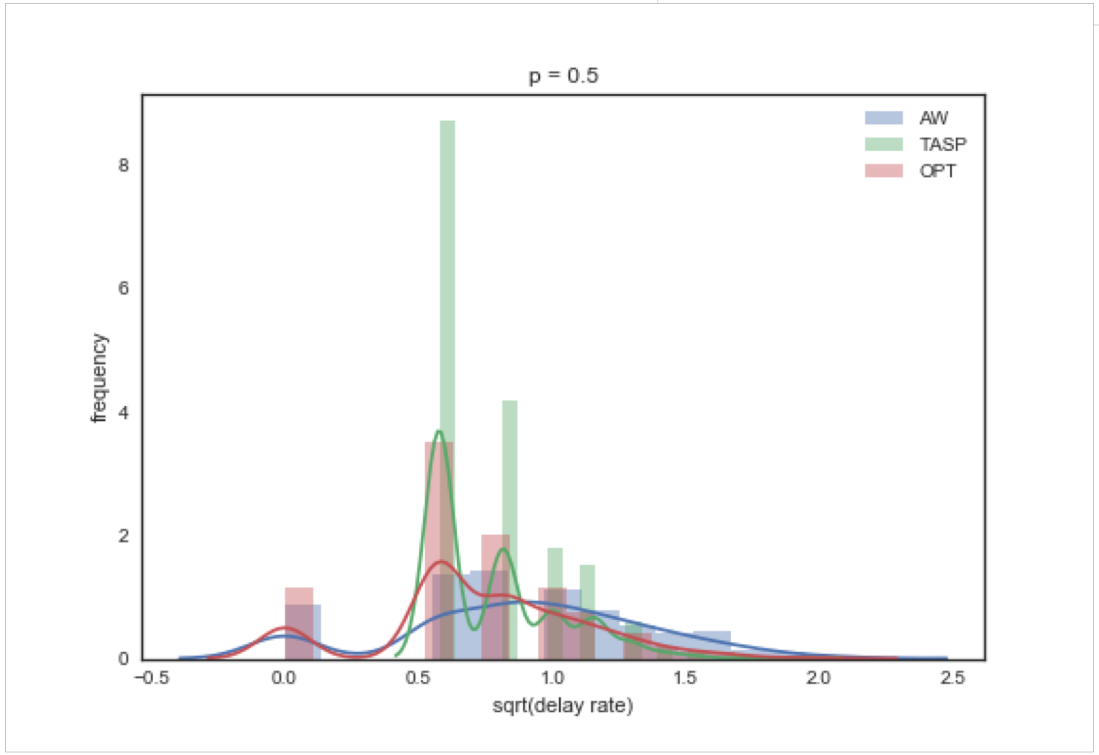


Simulation – bad case for AW Policy

- When p gets large, AW policy becomes better:
  - The advantage of the "broad way" is disappearing as p grows large







Bibliography

- P. Holme. Network reachability of real-world contact sequences.
- David Kempe, Jon Kleinberg, and Amit Kumar. Connectivity and inference problems for temporal networks. pages 504–513, 2000.

- Thanks to Professor Chris De Sa who suggested using stochastic dynamic programming

Thank you!