Spatially Distributed Queues

M/G/1

2 Servers

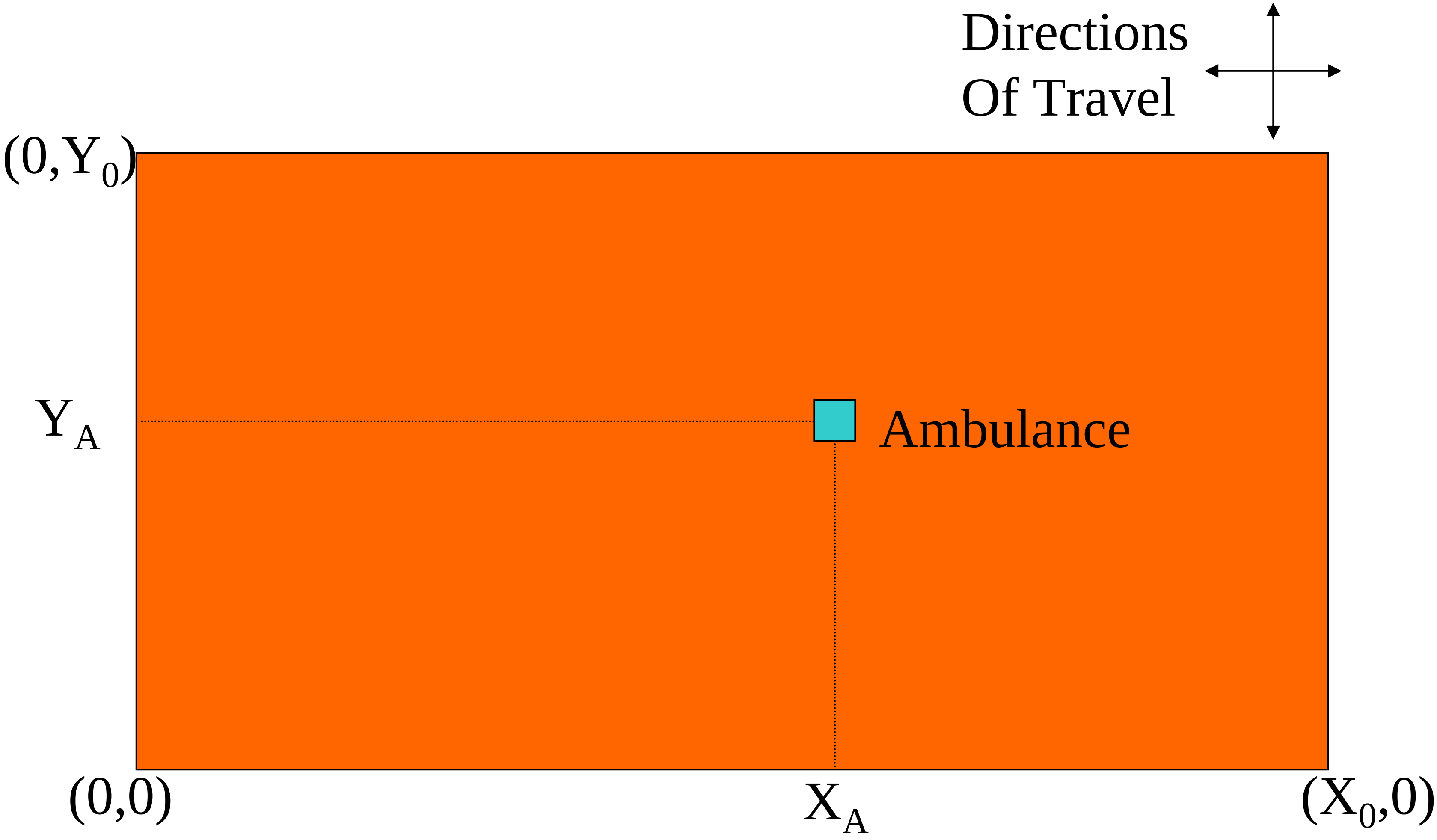
N servers

Approximations

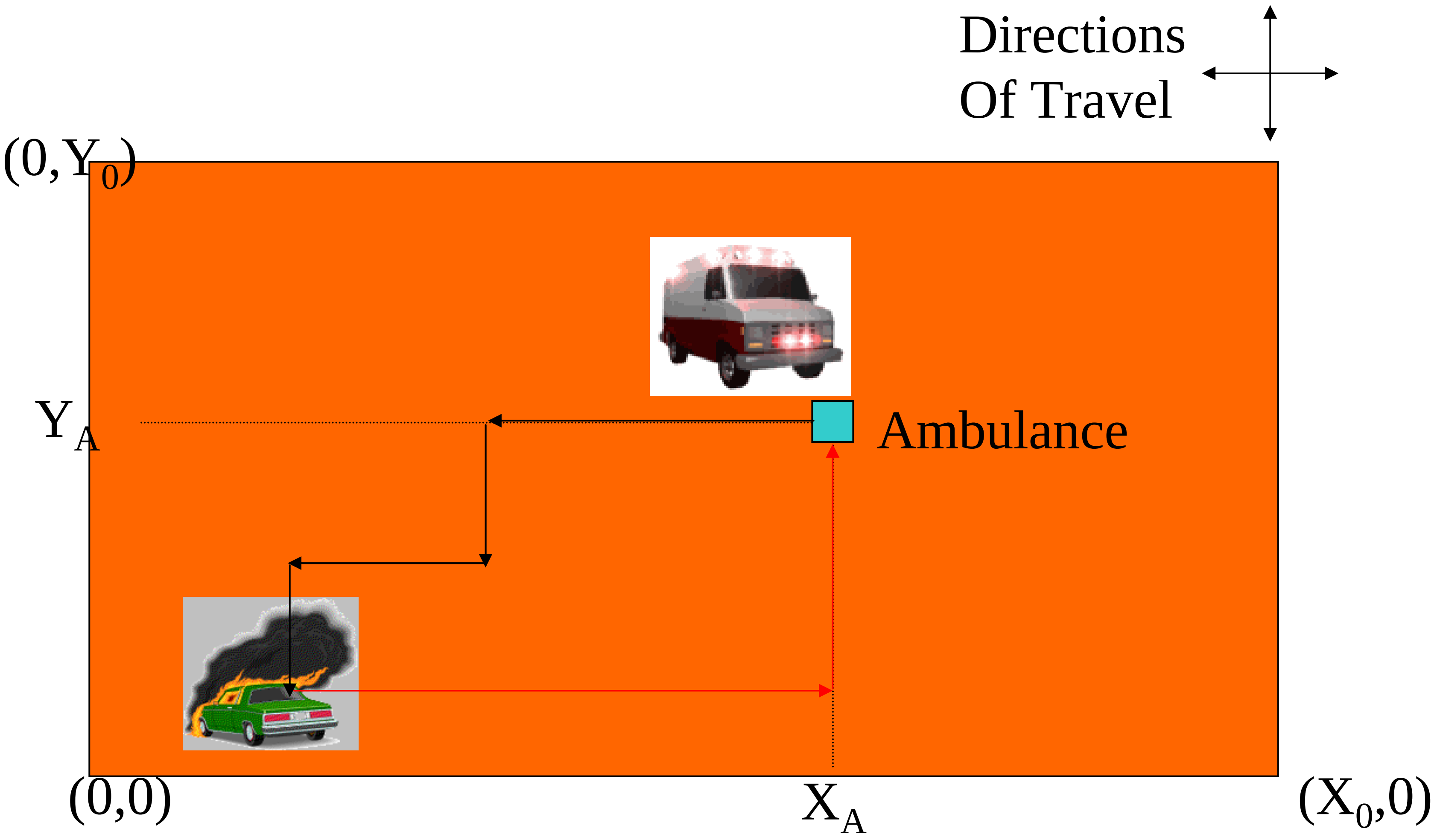
Why Spatial Queues?

- Demand responsive transportation systems
- Organ donation queues
- Warehouses
- Supply chains
- Cell phone systems
- People waiting to be evacuated in a hurricane

M/G/1



M/G/1



M/G/1

- Ambulance always returns home with each service; standard M/G/1 applies
- But suppose we have an emergency repair vehicle that travels directly from one customer to the next?……

M/G/1 with different 1st service time

$\overline{S}_1, \sigma^2_{S_1}$ = expected value and variance,respectively,

of the 1st service time in a busy period

$\overline{S}_2, \sigma^2_{S_2}$ = expected value and variance,respectively,

of the 2nd & all succeeding service times in a busy period

$\lambda \overline{S}_2 < 1$

$\rho = 1 - P_0$ = fraction of time server is busy

M/G/1 with different 1st service time

$$\rho = \frac{\lambda \overline{S_1}}{1 - \lambda(\overline{S_2} - \overline{S_1})}$$

$$L = \rho + \frac{\lambda^2}{1 - \lambda(\overline{S_2} - \overline{S_1})}\left[\frac{\sigma_{S_1}^2 + \overline{S}_1^2 + \lambda\{\overline{S}_1(\sigma_{S_2}^2 + \overline{S}_2^2) - \overline{S}_2(\sigma_{S_1}^2 + \overline{S}_1^2)}{2(1 - \lambda\overline{S}_2)}\right]$$

M/G/1 with different 1st service time

Little's Law: Buy one, get three others for free!

$$L = \lambda W$$

$$L_q = \lambda W_q$$

See the book, Eqs. (5.0) - (5.5)

M/G/1 with different 1st service time

- Does this new more general M/G/1 model apply exactly to the ambulance problem?

- Why or why not?

Two-Server "Hypercube" Queueing Model

- Distinguishable servers
- Different workloads (due to geography)
- Can appear with or without queueing
  - With -- usually FCFS
  - Without -- usually means a backup contract service is in place


119
瀬戸市
消防本部


FIRE DEPARTMENT
FIRE DEPARTMENT


11
瀬戸市
消防本


119
瀬戸市
消防本部

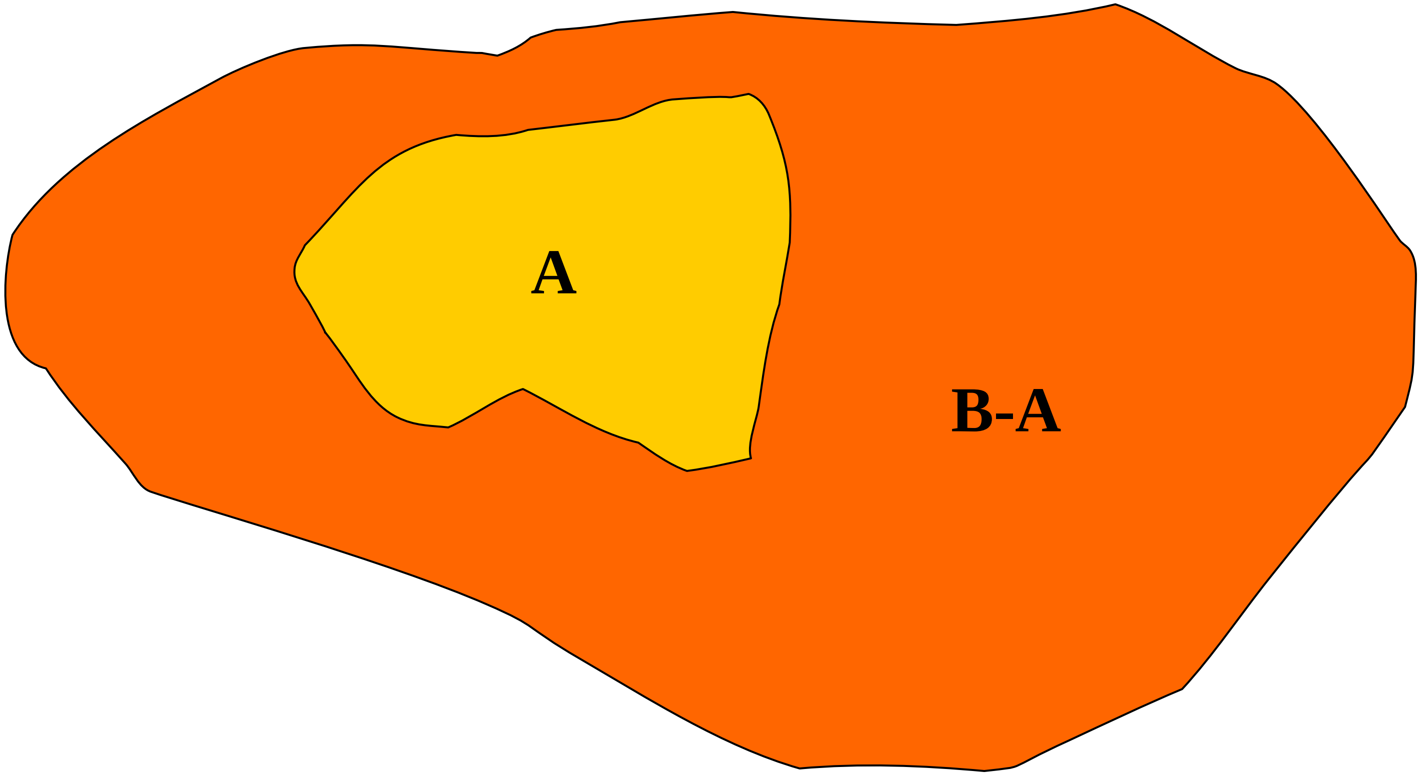

119
瀬戸市
消防本部


FIRE DEPARTMENT
FIRE DEPARTMENT


119
瀬戸市
消防本部

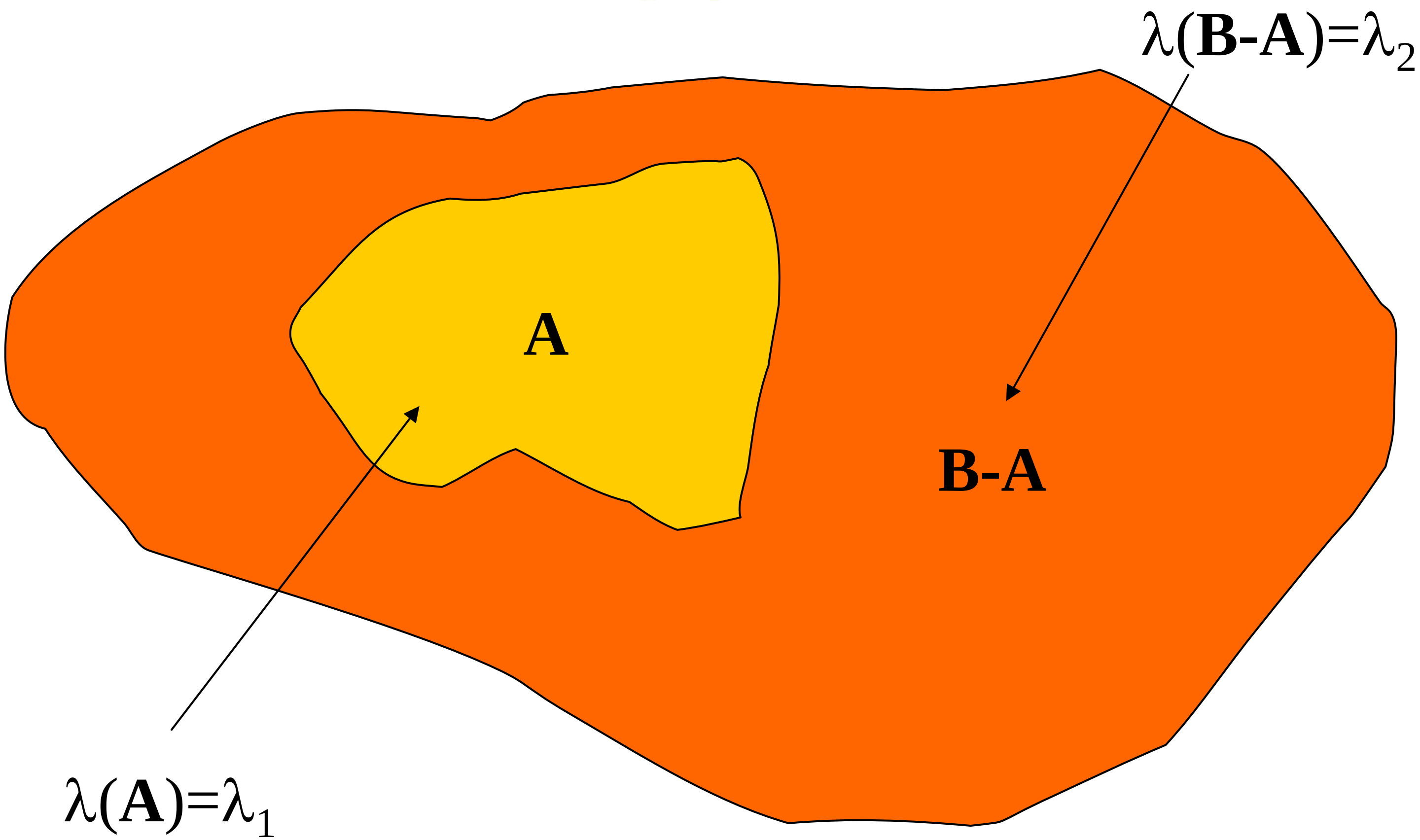

O.R.
SCIENCE
BETTER


O.R.
SCIENCE
BETTER



B = "Service Region"

Poisson Arrivals from any sub-region **A**

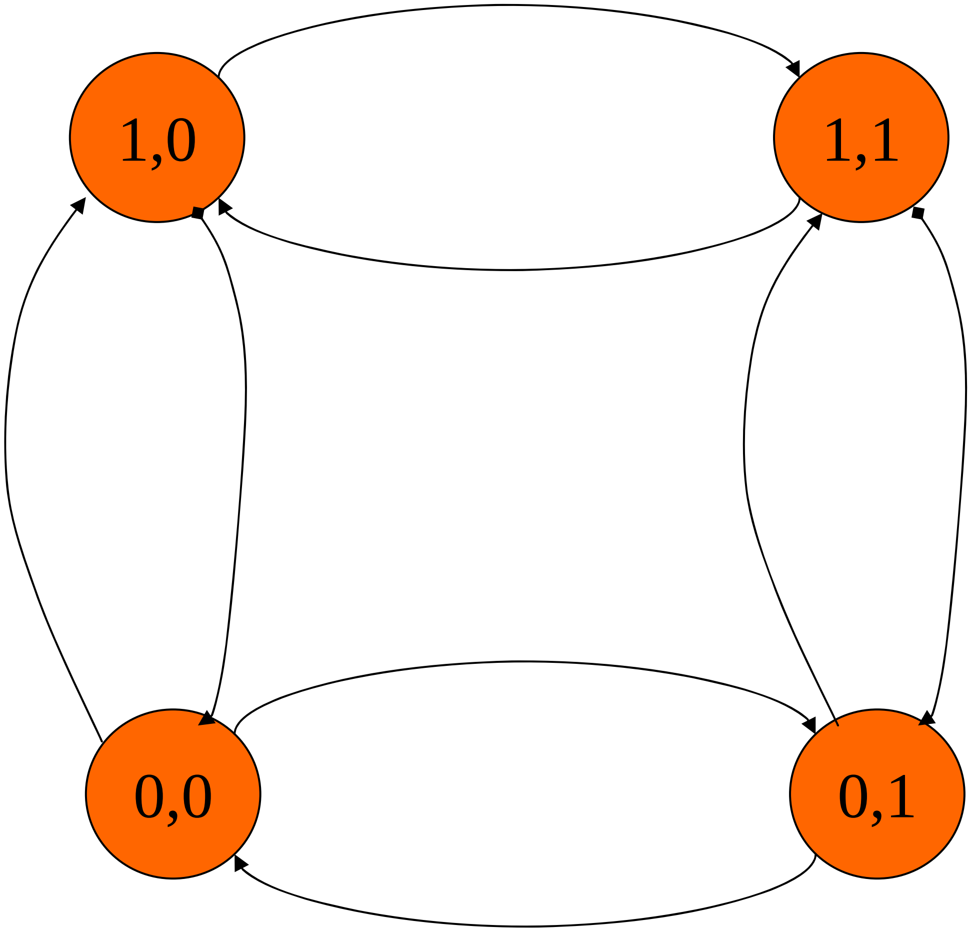


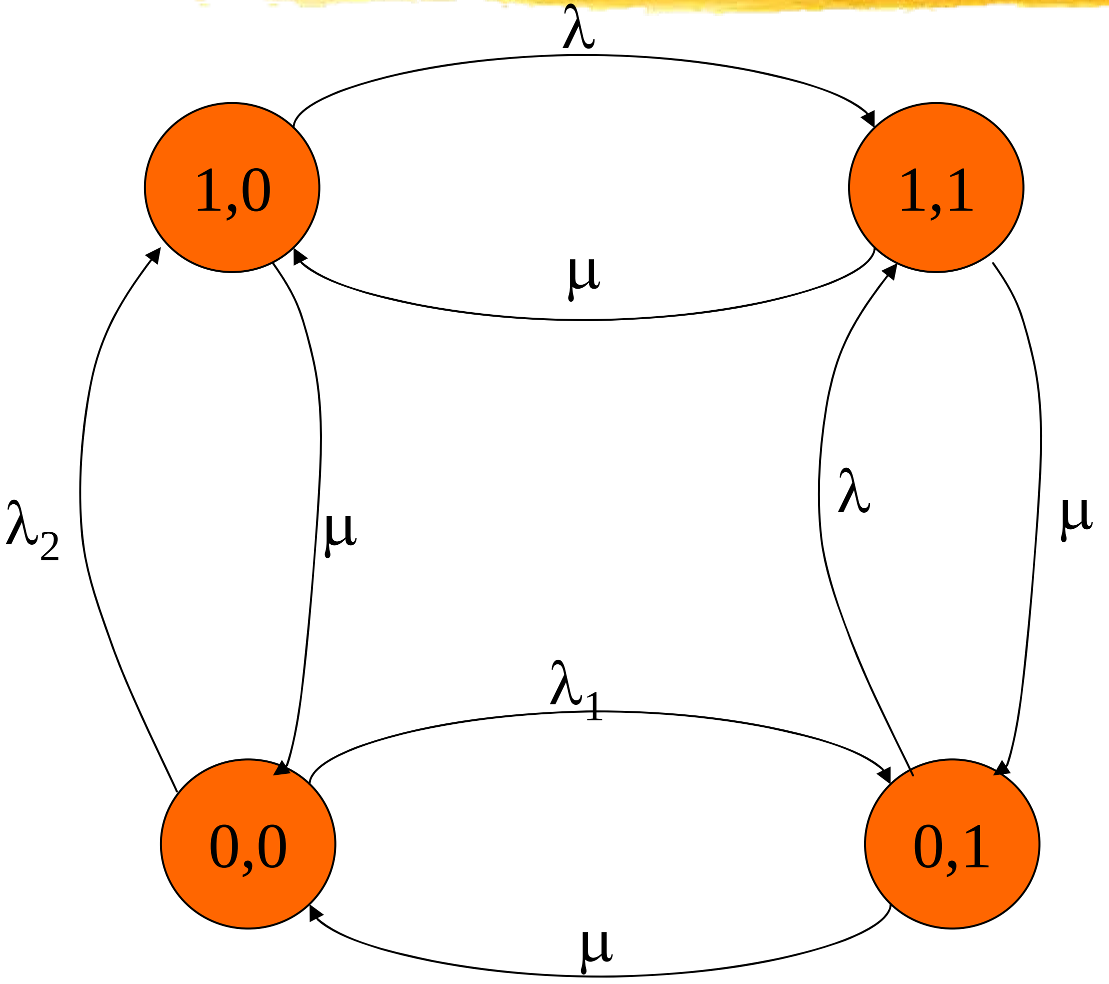
B = "Service Region"

$\lambda = \lambda_1 + \lambda_2$

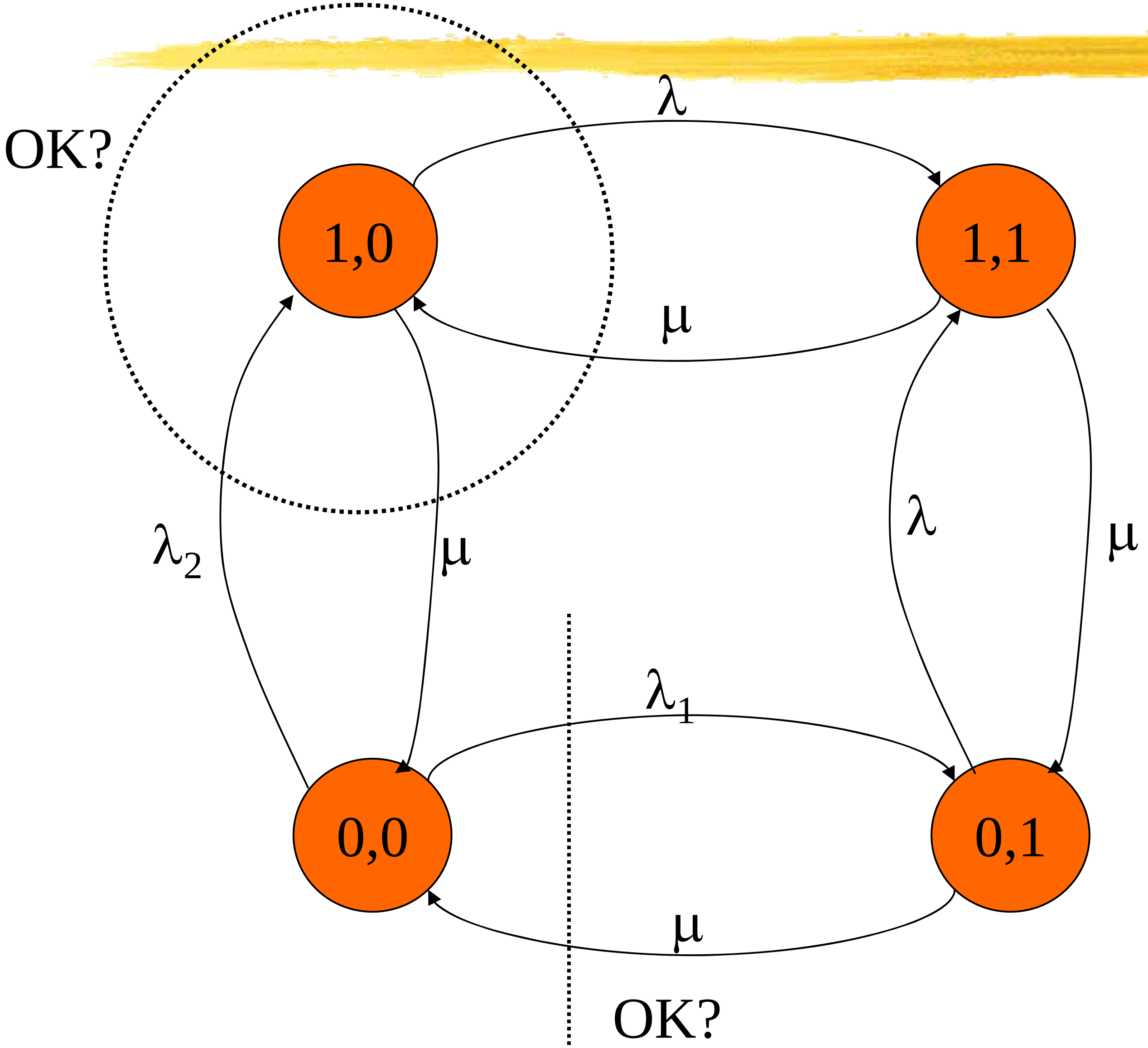
Service Discipline

- 1st Dispatch Preference to ‘primary server’
- Otherwise, assign customer to other server, if available
- Otherwise, job is ‘lost” (What happens in practice?)

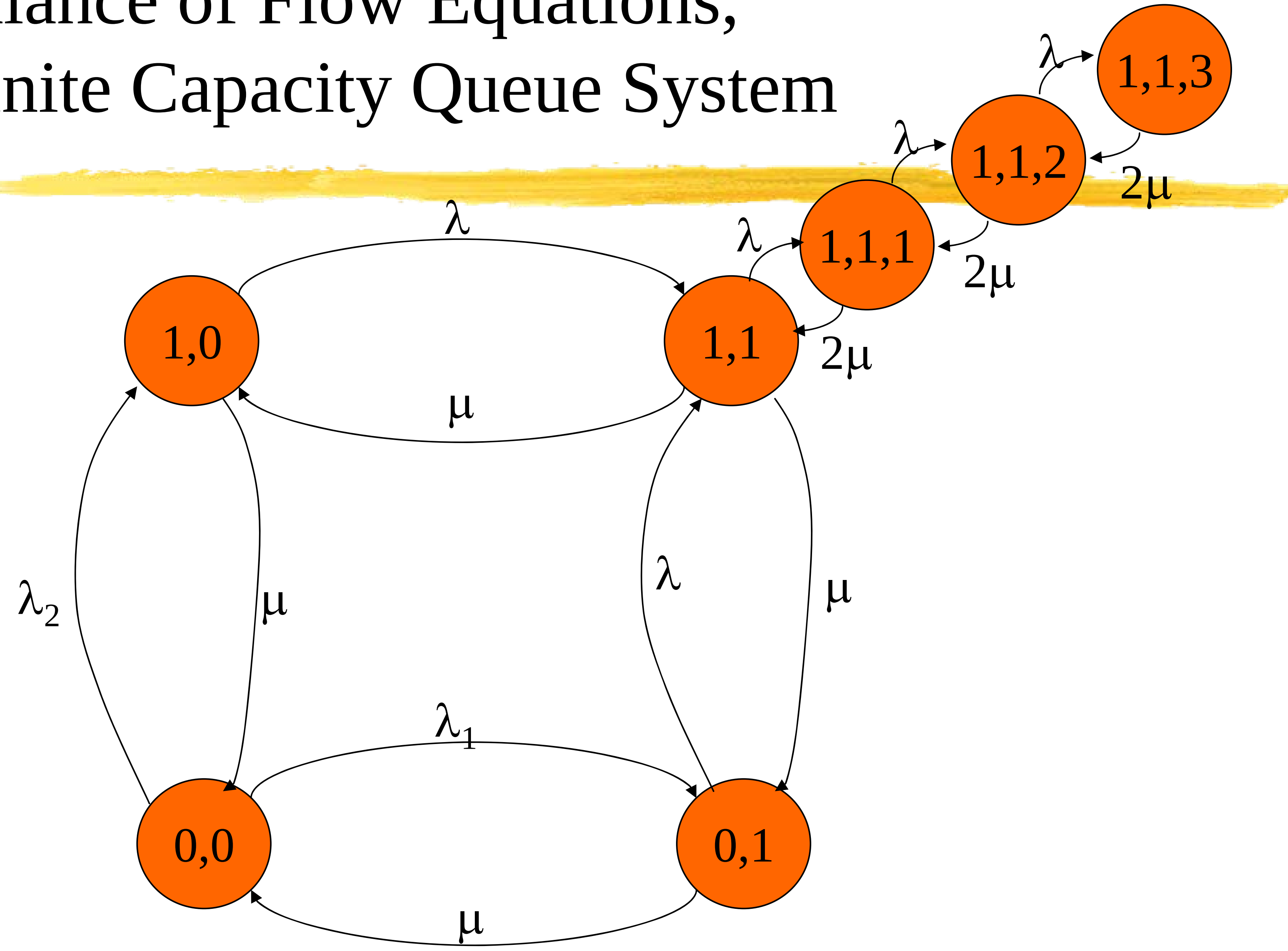

1,0
1,1
0,0
0,1


λ
1,0
1,1
μ
λ2
μ
λ
μ
λ1
0,0
0,1
μ

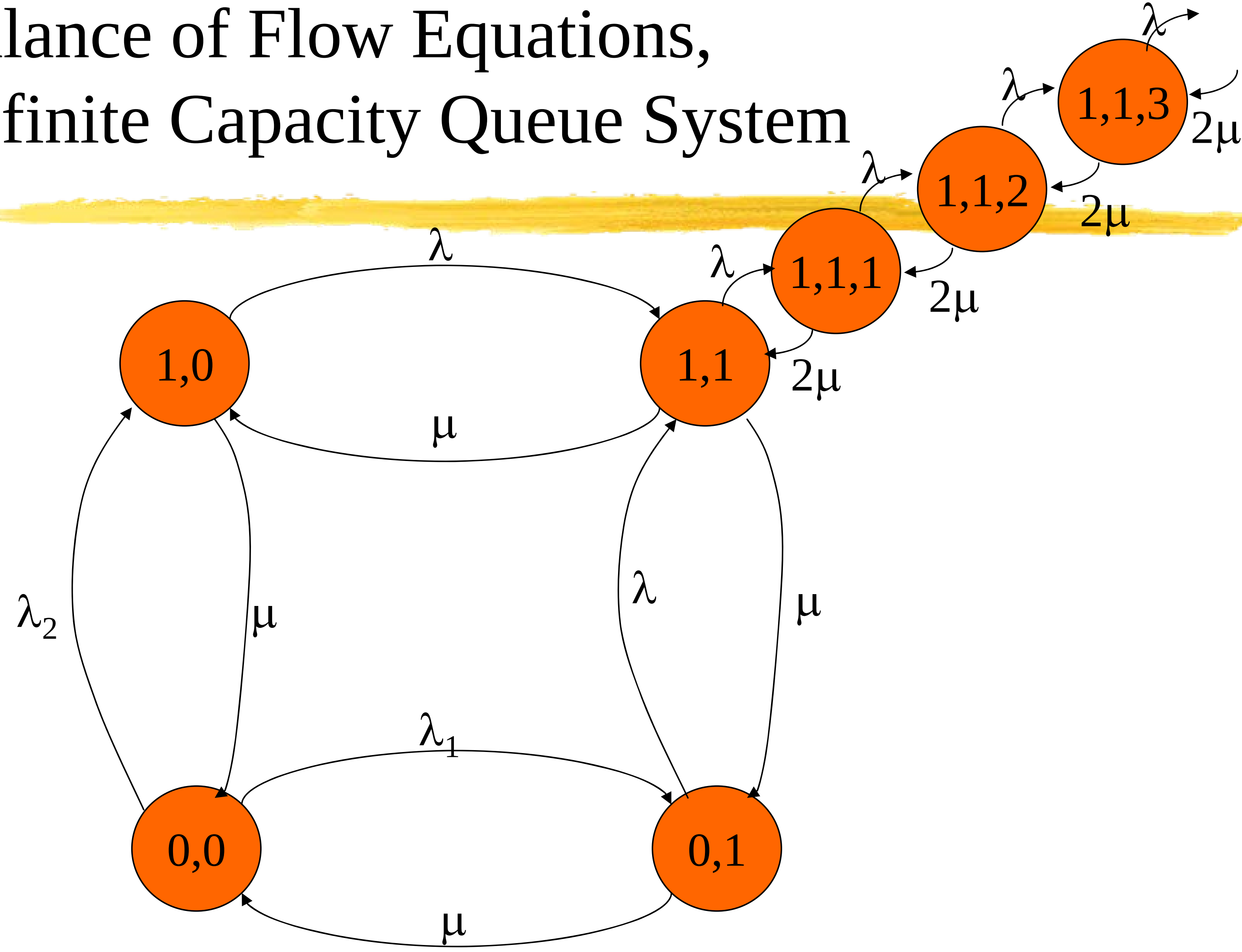
Balance of Flow Equations, Loss System



Balance of Flow Equations, Finite Capacity Queue System



Balance of Flow Equations, Infinite Capacity Queue System



Balance of Flow Equations, Loss System

$$P_{00}\,(\lambda_1 + \lambda_2) = P_{01}\mu + P_{10}\mu$$

$$P_{01}\,(\lambda + \mu) = P_{11}\mu + P_{00}\,\lambda_1$$

Etc.

$$P_{00} + P_{10} + P_{01} + P_{11} = 1$$

Workload and Imbalances

- $\rho_1 = W_1 = P_{01} + P_{11}$
- $\rho_2 = W_2 = P_{10} + P_{11}$
- *Workload Imbalance* $= \Delta W = |W_1 - W_2|$

To Obtain Travel Times, We Must Have Server Response Patterns

- ⌘ f_{nj} = fraction of dispatches that are server n to response area j
- ⌘ $T_n(\mathbf{C})$ = average time for server n to travel to a customer in region **C**
- ⌘ $T(\mathbf{A})$ = average system-wide travel time, assuming that server 1's primary response area is region **A**.

Average System-Wide Travel Time

$$T(\mathbf{A}) = f_{11} T_1(\mathbf{A}) + f_{22} T_2(\mathbf{B\text{-}A}) + f_{12} T_1(\mathbf{B\text{-}A}) + f_{21} T_2(\mathbf{A})$$

Average System-Wide Travel Time

$$T(\mathbf{A}) = f_{11} T_1(\mathbf{A}) + f_{22} T_2(\mathbf{B\text{-}A}) + f_{12} T_1(\mathbf{B\text{-}A}) + f_{21} T_2(\mathbf{A})$$

Queueing

Average System-Wide Travel Time

$$T(\mathbf{A}) = f_{11} T_1(\mathbf{A}) + f_{22} T_2(\mathbf{B\text{-}A}) + f_{12} T_1(\mathbf{B\text{-}A}) + f_{21} T_2(\mathbf{A})$$

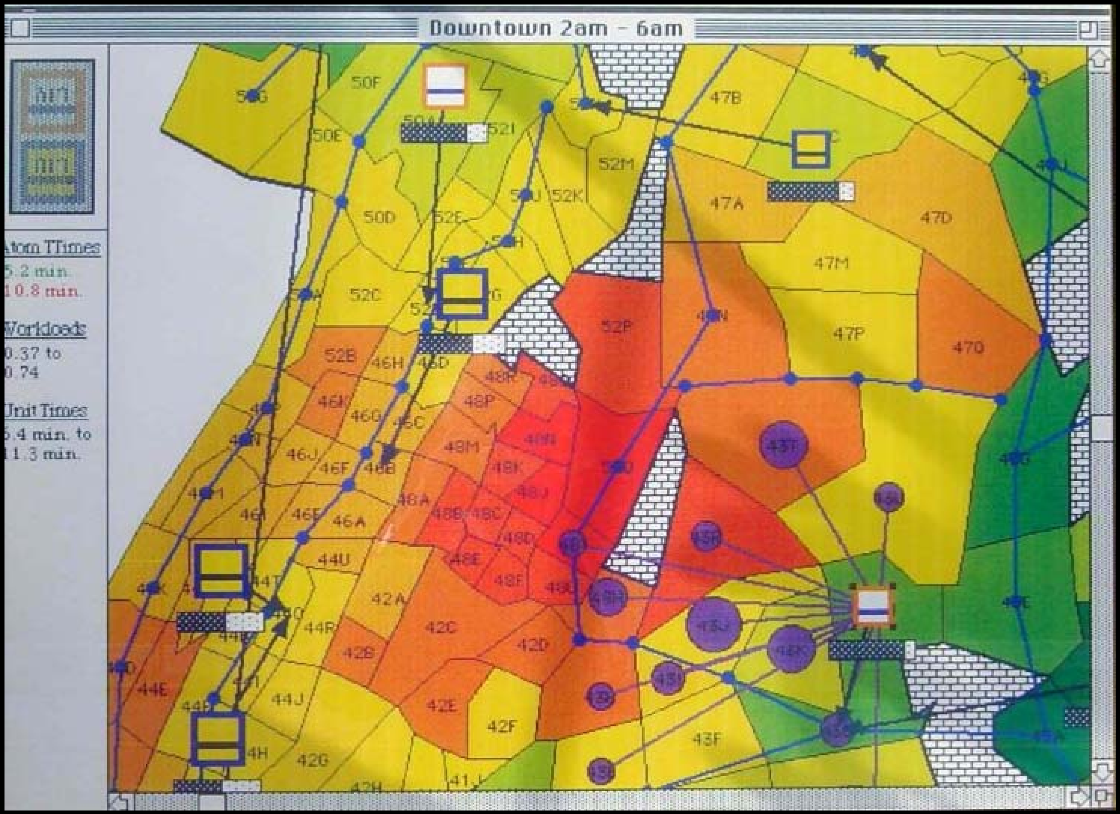
Geometry

How do we obtain the f_{nj}'s?

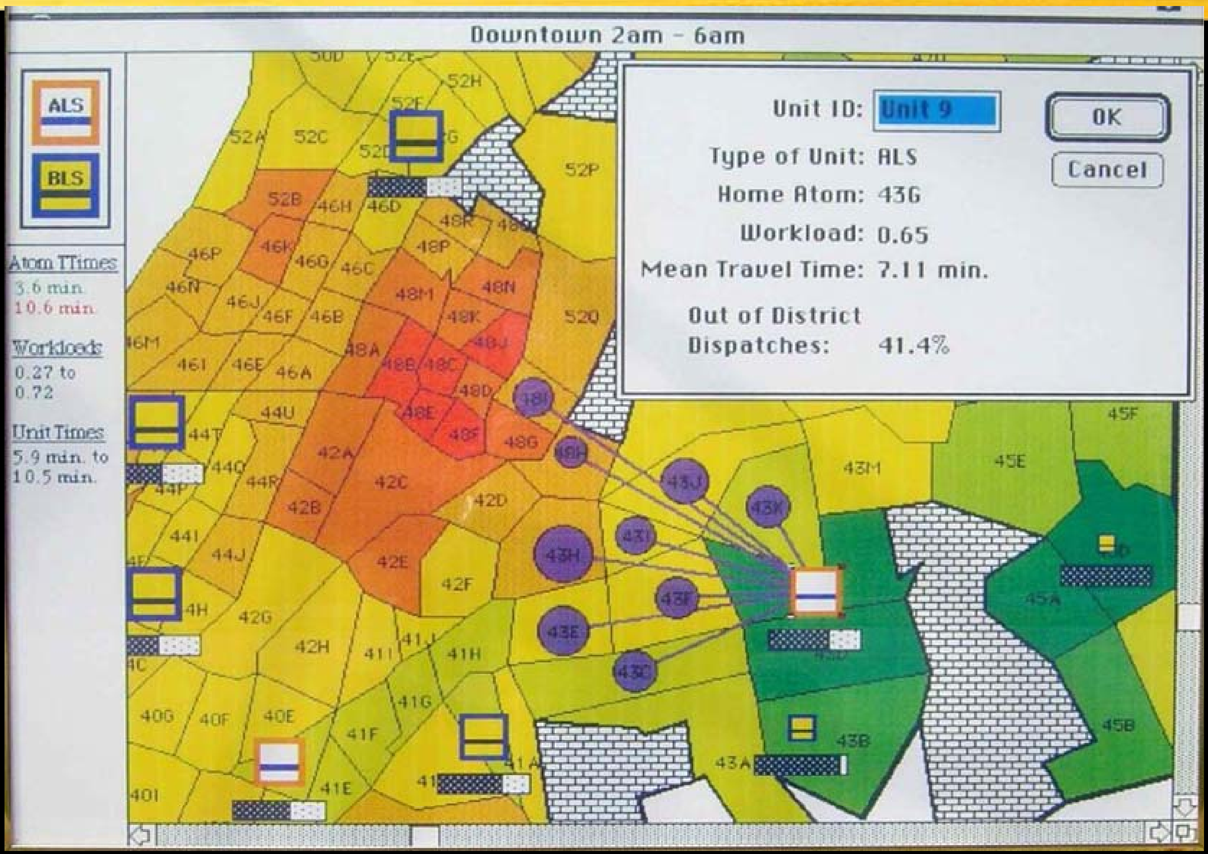
- Consider a long time interval T
- f_{12}=(# requests that assign unit 1 to area 2)/ (total # requests answered)
- Total # requests answered = $(1-P_{11})\lambda T$
- Average # requests that are "server 1 to area 2" is $\lambda_2 T P_{10}$. Why?
- Therefore $f_{12} = (\lambda_2 T P_{10} / [1-P_{11}]\lambda T) = \{\lambda_2 / ((1-P_{11})\lambda)\} P_{10}$

How do we generalize this to *N* servers?

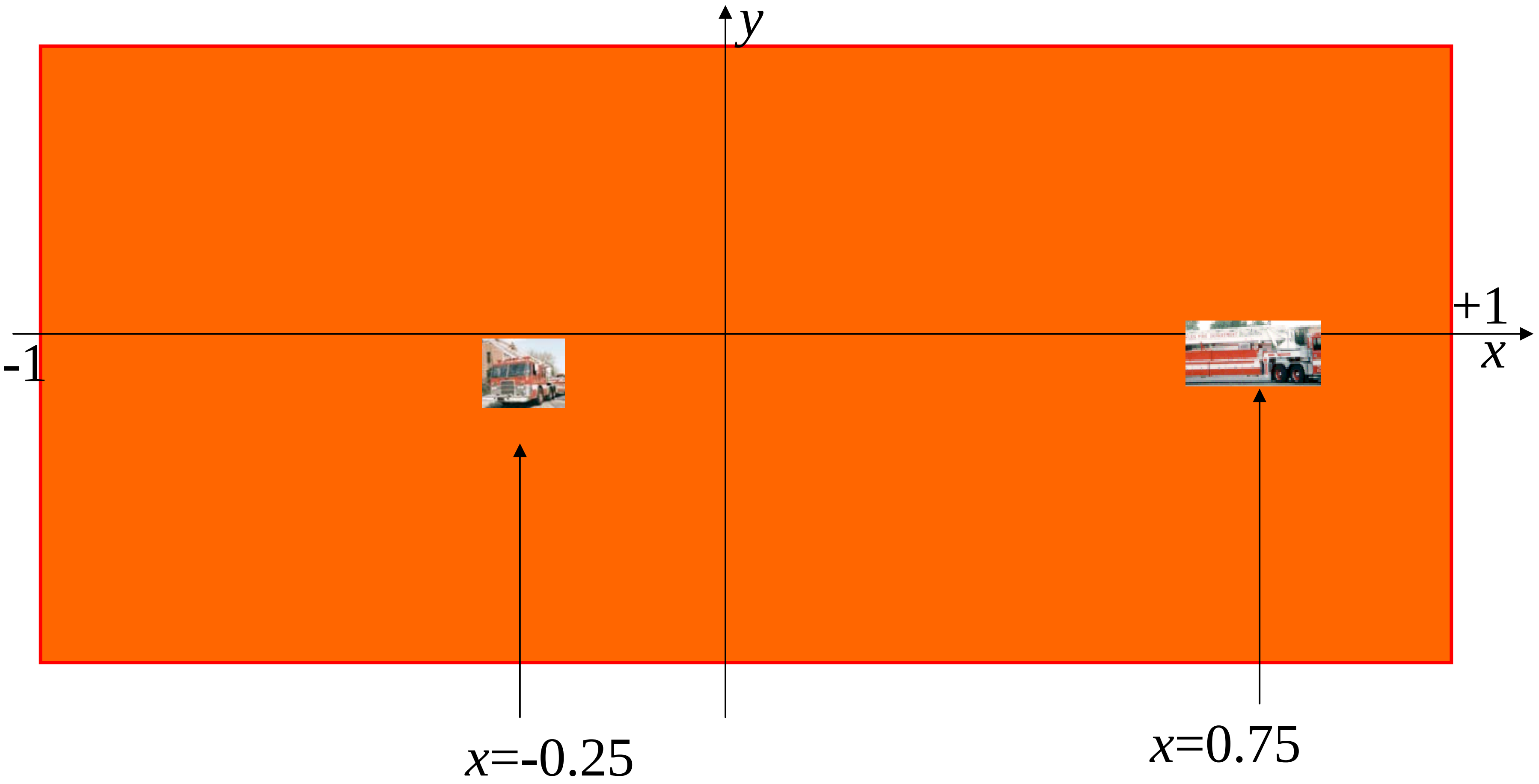
New York City
EMS Hypercube



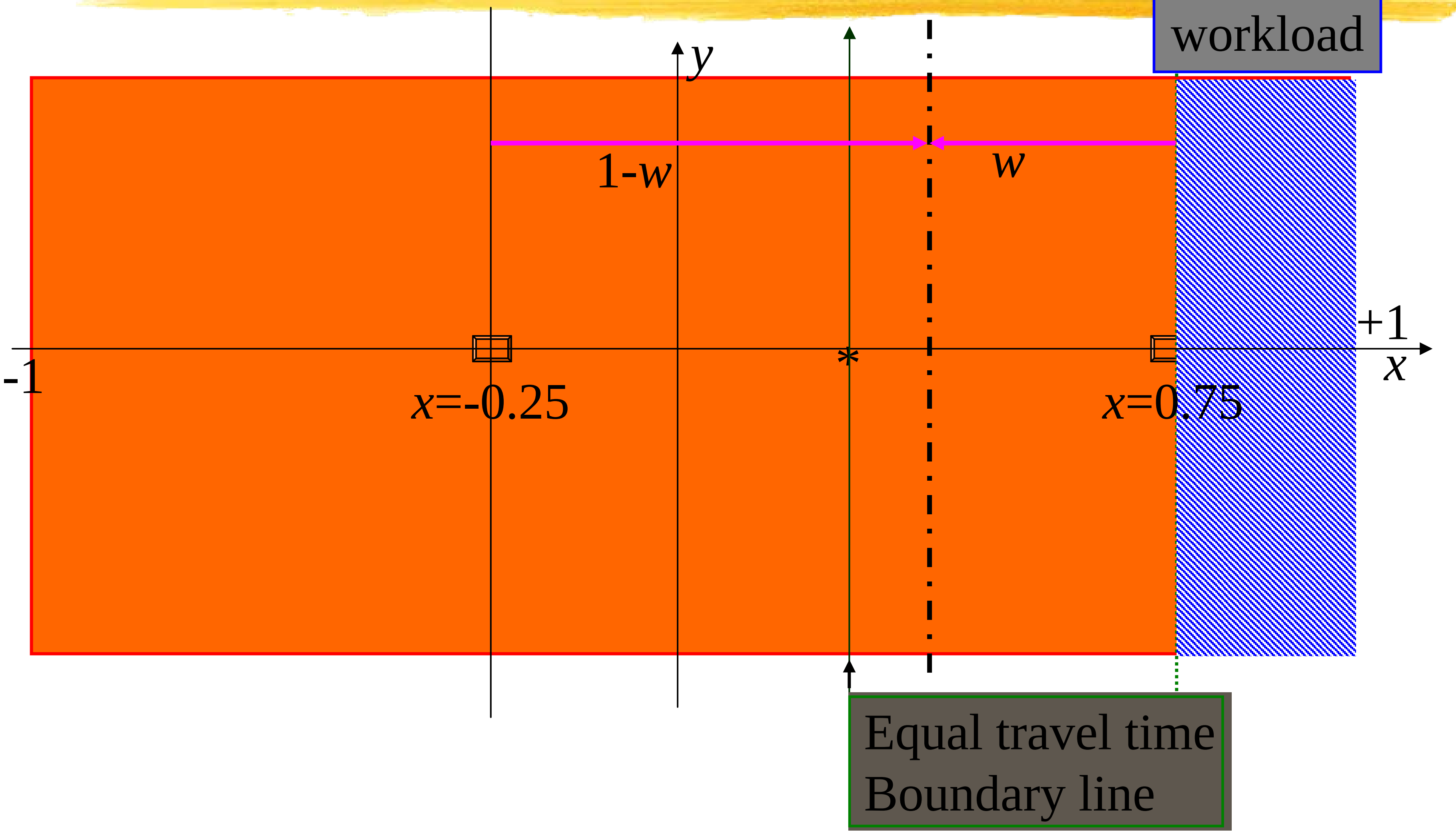
New York City EMS Hypercube



Rectangular City Example



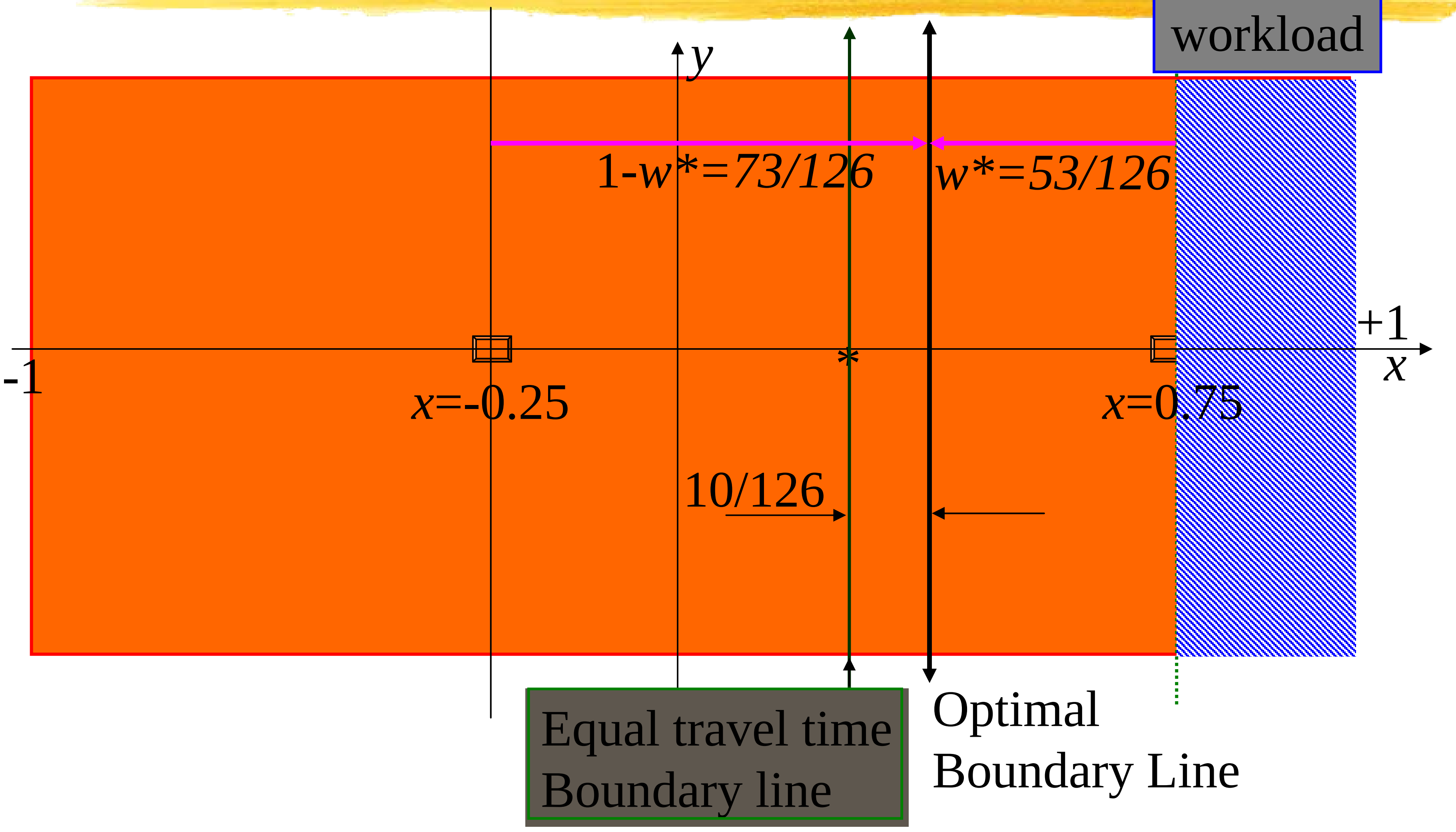
Rectangular City Example



Optimal Districting

- “Dispatch the closest available server’ is often not optimal, where ‘optimal’ implies minimizing mean travel time
- May not be good for reducing workload imbalance either
- With numerical example in book, the optimal boundary line is shifted to the right by 10/126 miles.

Rectangular City Example



Boundary Line Comparison

- Equal travel time boundary line
  - $T(A_{w=1/2})=0.46246$
  - $\Delta W = 0.05236$
- Optimal boundary line
  - $T(A_{w^*})=0.46166$
  - $\Delta W = 0.04405$

Two server Loss Model: Boundary Line Result

- **To minimize mean city-wide mean travel time:**
- ***The optimal partitioning consists of a set of points within the region that is a constant travel time s_0 closer to facility 1 than to facility 2.*** (Carter, Chaiken, Ignall, 1972)
- Does our rectangular city example work for this?

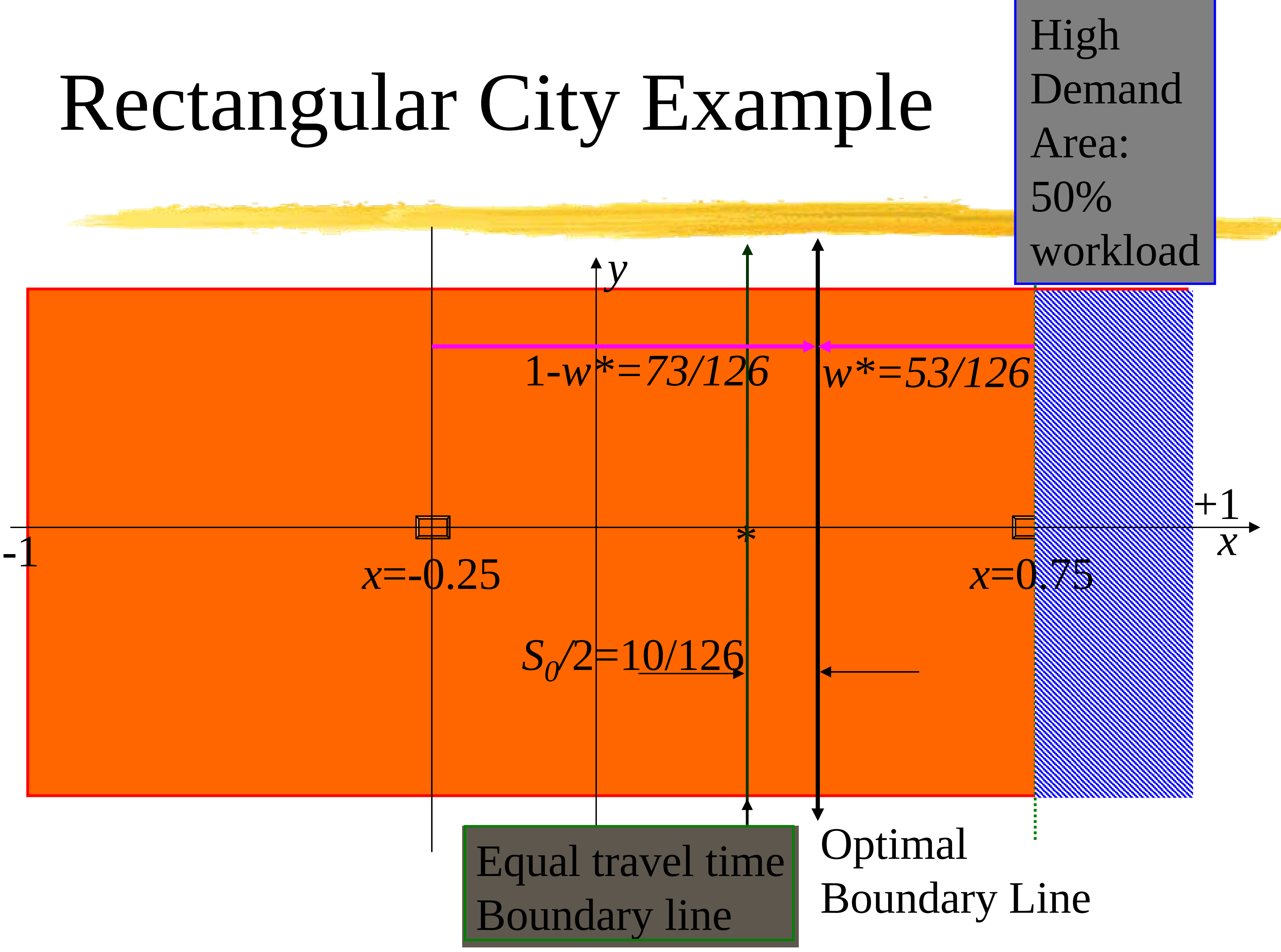
S_0: Optimal Partitioning

$$\alpha \equiv \lambda / 2\mu$$

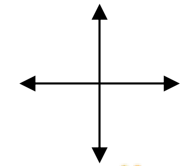
$$\mu_1 = \mu_2$$

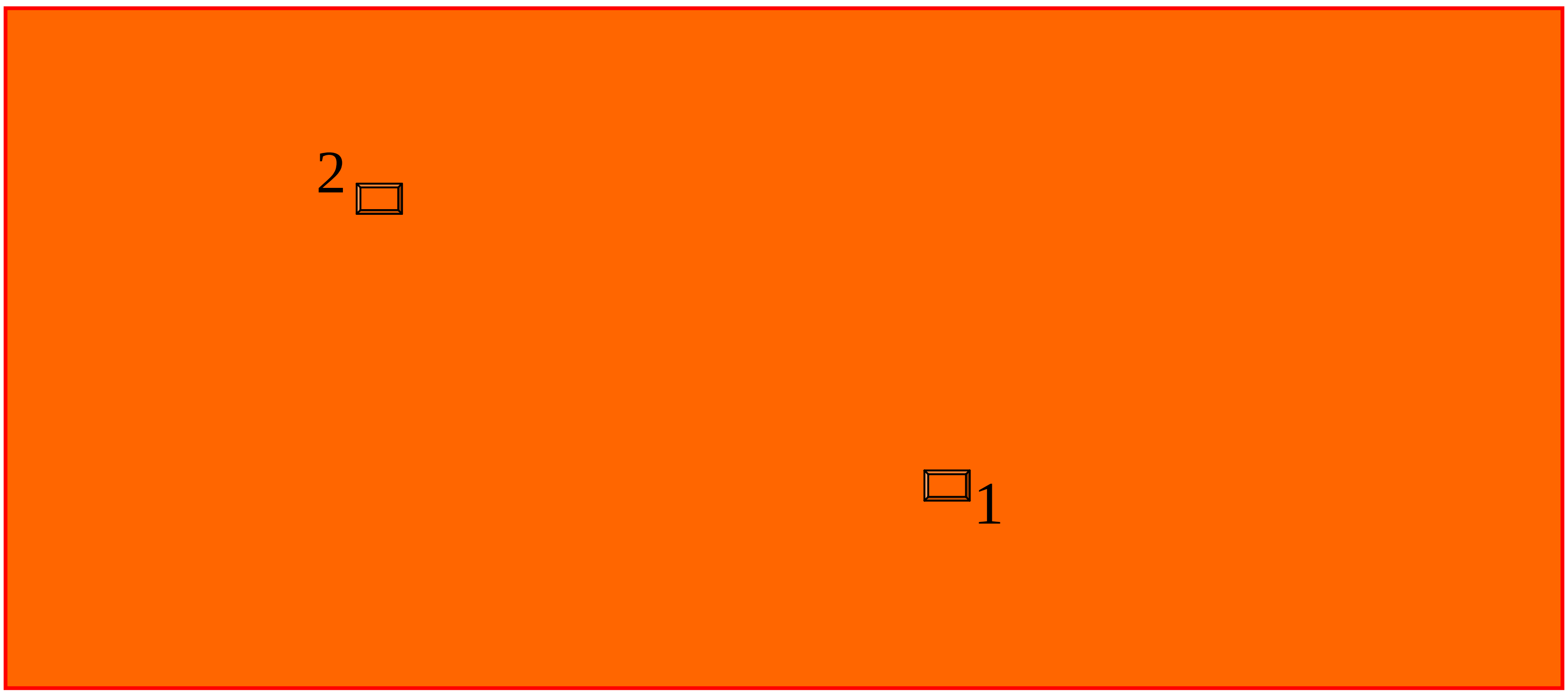
$$S_0 = [2\alpha /(2\alpha + 1)]\{T_2(B) - T_1(B)\}$$

Rectangular City Example

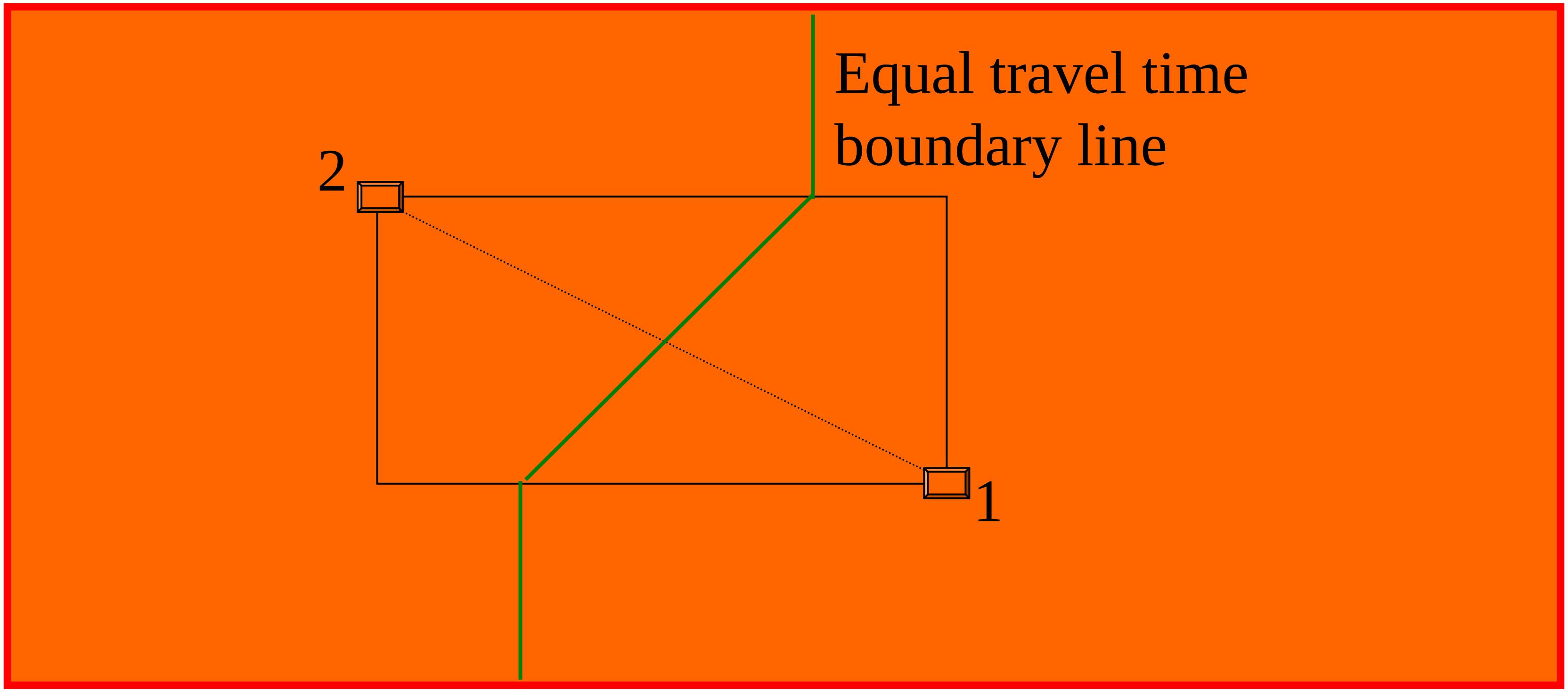


Directions
Of Travel

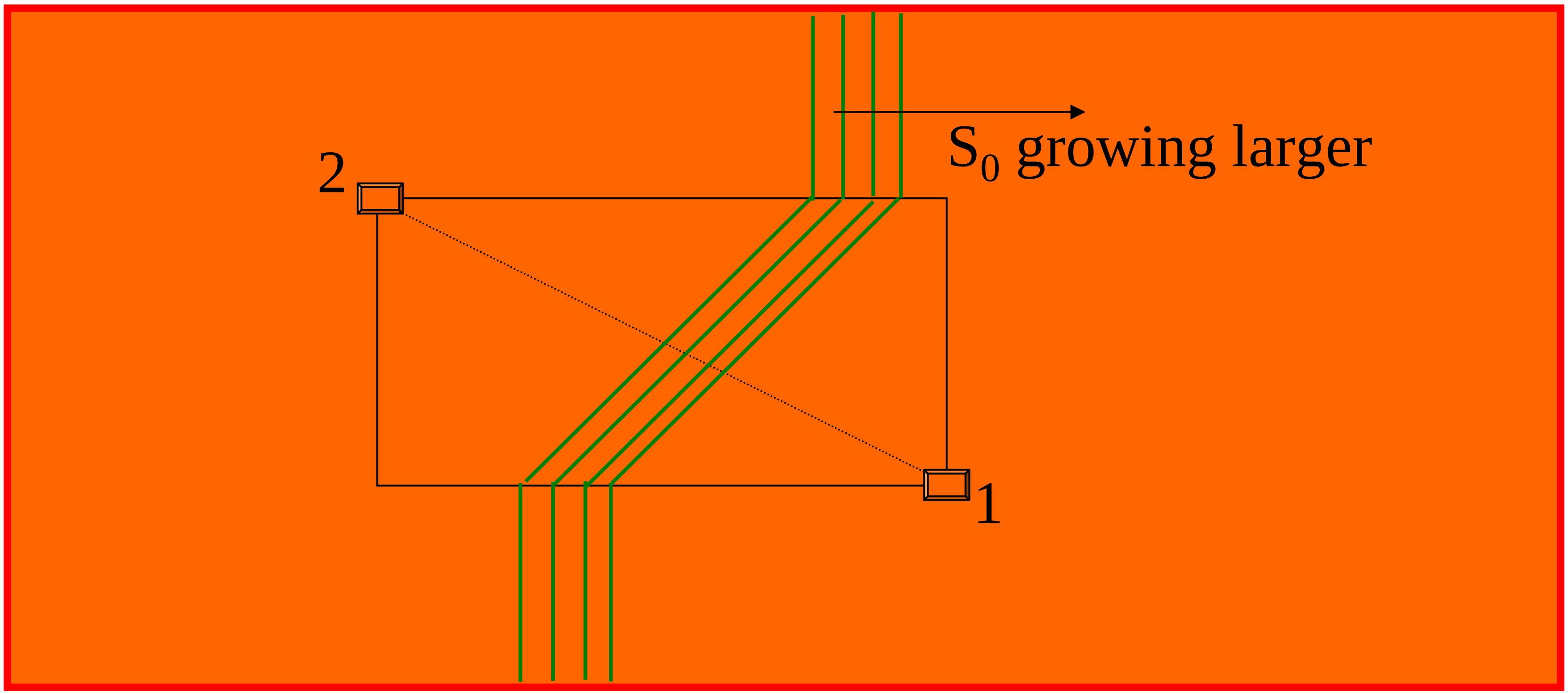




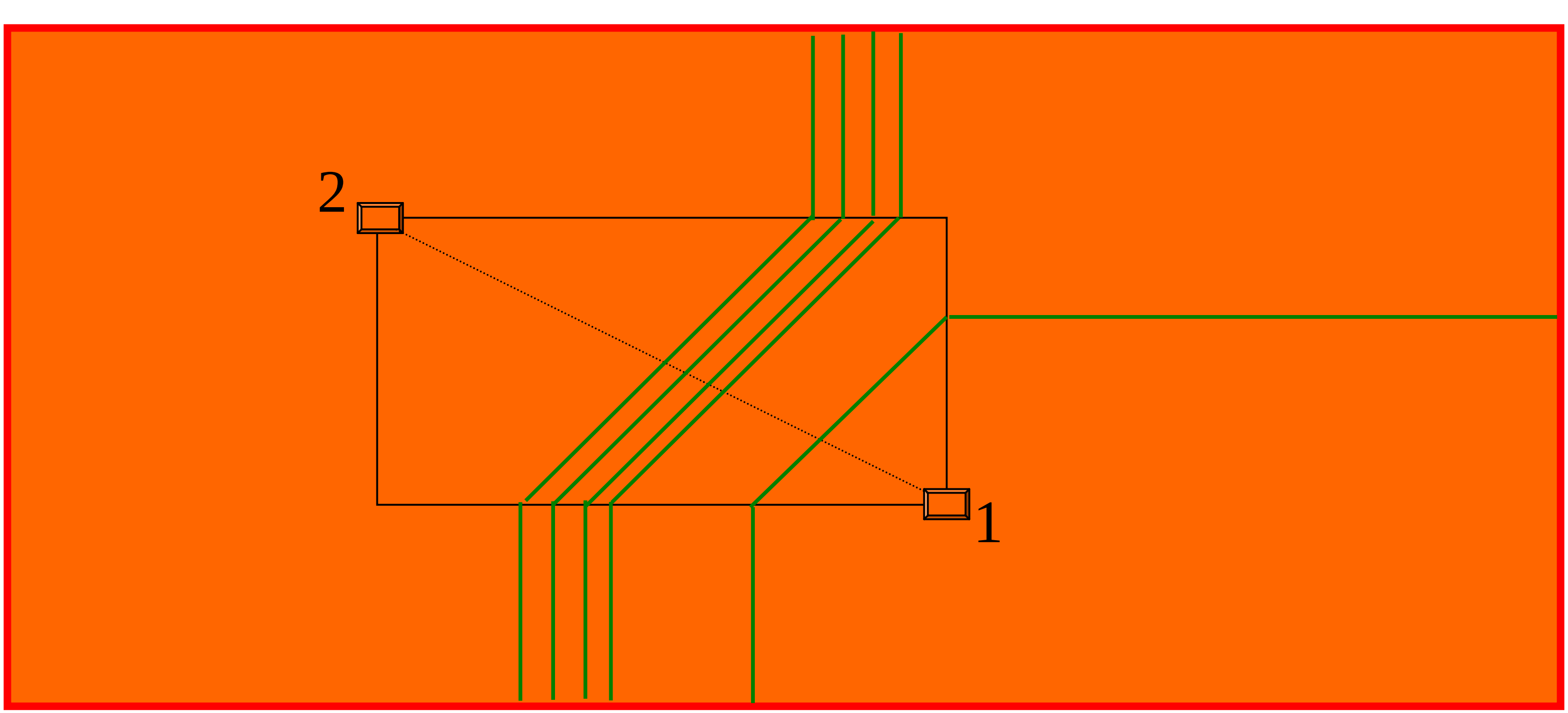
Directions
Of Travel



Directions Of Travel

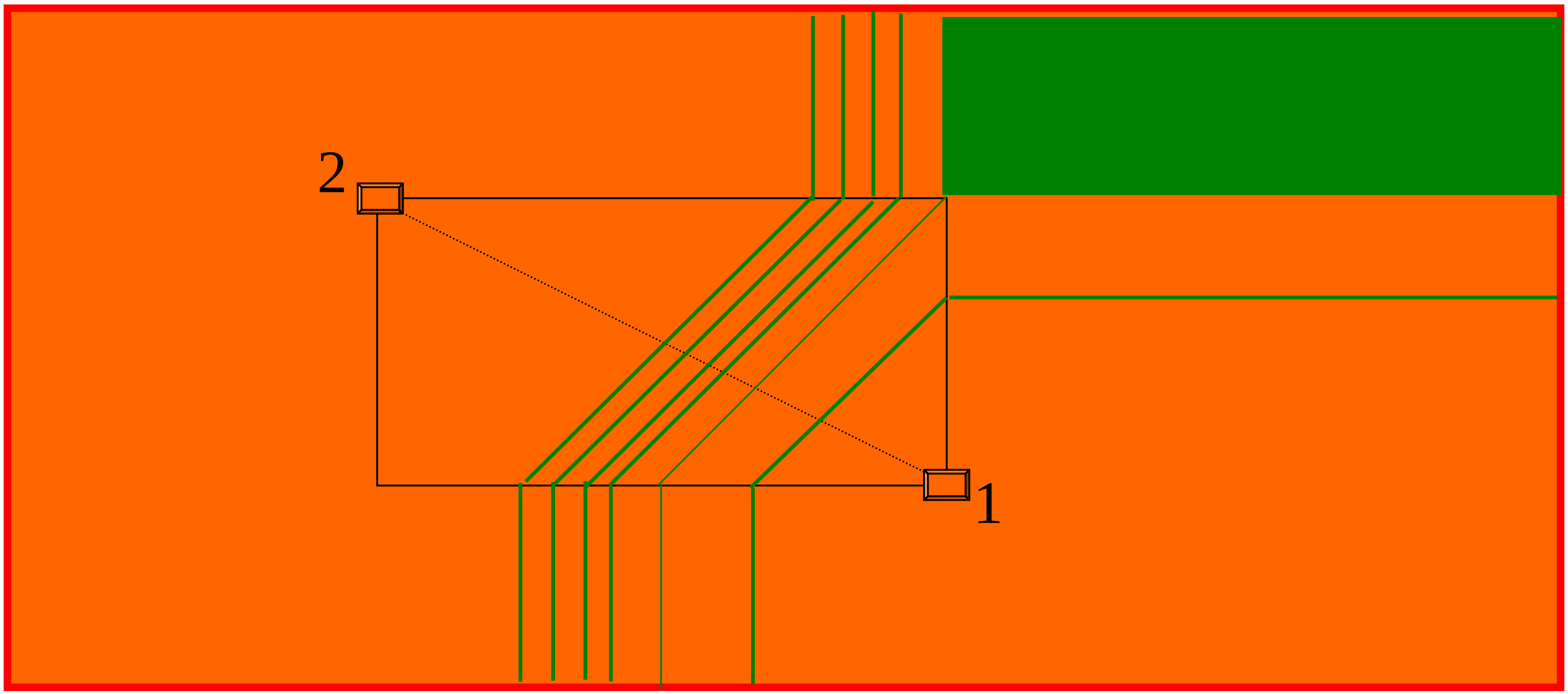



Directions
Of Travel


2
1

And what about the corner case?

Directions
Of Travel



The S_0 result is general

- Works for discrete grid
- One way streets
- General transportation network

- Rick Jarvis, in an MIT Ph.D. thesis, generalized this to N servers