Spatial Inefficiency of MaxWeight Scheduling

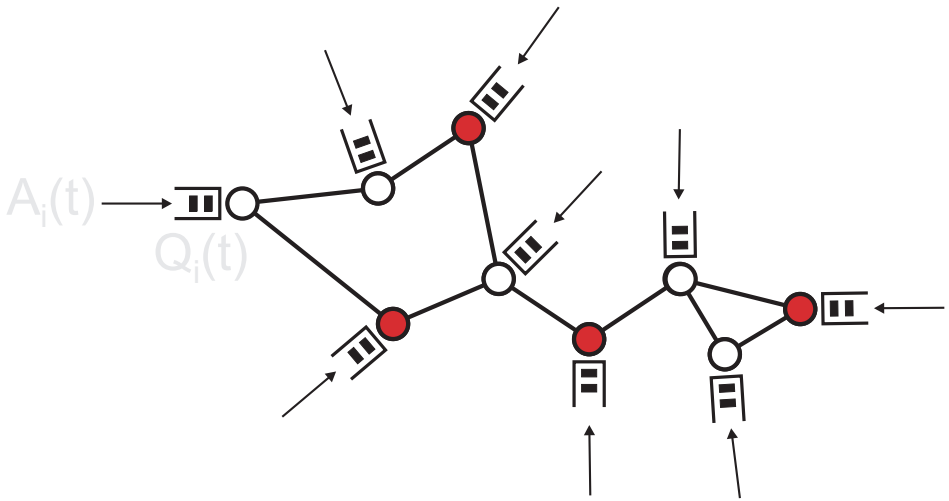
Peter van de Ven

joint work with

Sem Borst (TU/e and Bell Labs) and Lei Ying (Iowa State University)

YEQT 2011
October 24, 2011

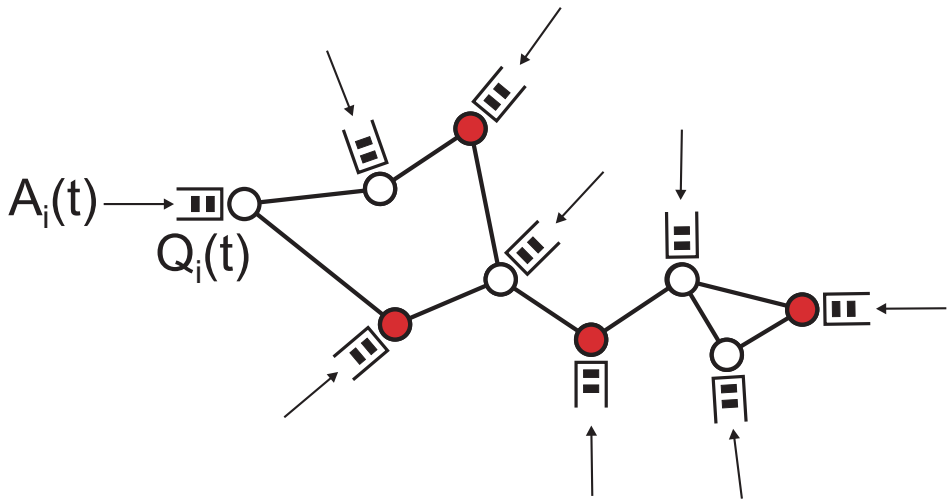
Wireless scheduling



Consider wireless scheduling under interference constraints:

- time is slotted
- flows (queues) that contain some packets (customers)
- the interference constraints are given by an interference graph
- set of feasible schedules Ω

Network dynamics



- each flow has i.i.d. arrivals of packets $A_i(t)$
- denote by $Q_i(t)$ the backlog of flow i at the start of slot t
- select in each time slot some feasible subset of flows $J(t) \in \Omega$
- scheduled flows have one packet served:

$$Q_i(t+1) = \left(Q_i(t) + A_i(t) - \mathbf{1}_{\{i \in J(t)\}}\right)^+$$

How to choose the set $J(t) \in \Omega$ for stability?

How to choose the set $J(t) \in \Omega$ for stability?

MaxWeight scheduling selects a feasible subset of flows with maximum aggregate backlog

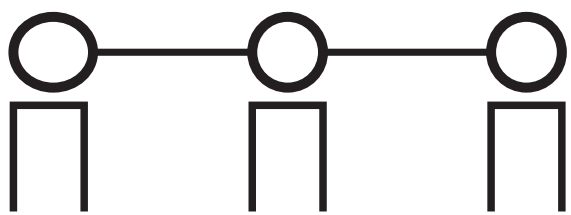
$$J^*(t) \in \arg\max_{J \in \Omega} \sum_{j \in J} Q_j(t)$$

This choice is throughput-optimal [Tassiulas & Ephemides (1992)]

MaxWeight scheduling

For example:

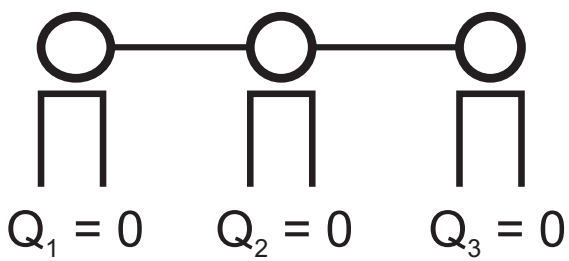
- linear network of 3 flows
- feasible schedules: $\Omega = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 3\}\}$
- MaxWeight scheduling compares $Q_2(t)$ and $Q_1(t) + Q_3(t)$



MaxWeight scheduling

For example:

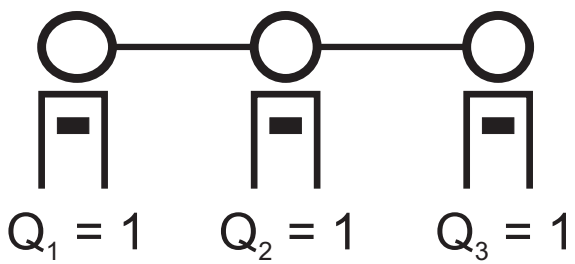
- linear network of 3 flows
- feasible schedules: $\Omega = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 3\}\}$
- MaxWeight scheduling compares $Q_2(t)$ and $Q_1(t) + Q_3(t)$



MaxWeight scheduling

For example:

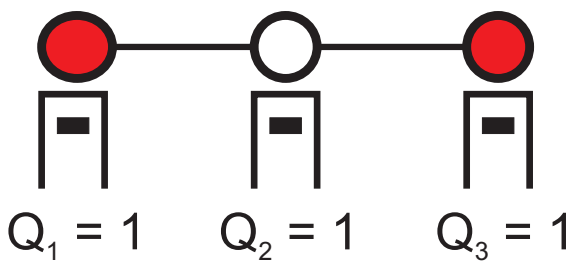
- linear network of 3 flows
- feasible schedules: $\Omega = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 3\}\}$
- MaxWeight scheduling compares $Q_2(t)$ and $Q_1(t) + Q_3(t)$



MaxWeight scheduling

For example:

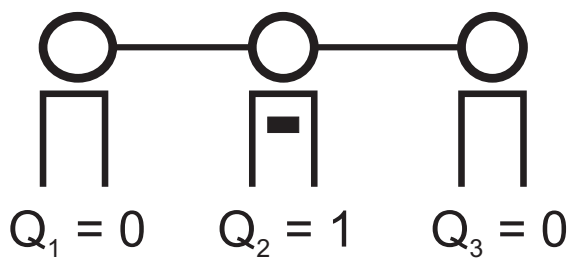
- linear network of 3 flows
- feasible schedules: $\Omega = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1,3\}\}$
- MaxWeight scheduling compares $Q_2(t)$ and $Q_1(t) + Q_3(t)$



MaxWeight scheduling

For example:

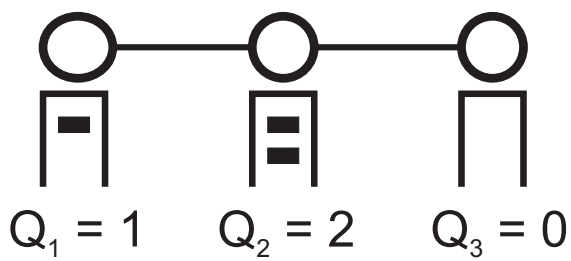
- linear network of 3 flows
- feasible schedules: $\Omega = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 3\}\}$
- MaxWeight scheduling compares $Q_2(t)$ and $Q_1(t) + Q_3(t)$



MaxWeight scheduling

For example:

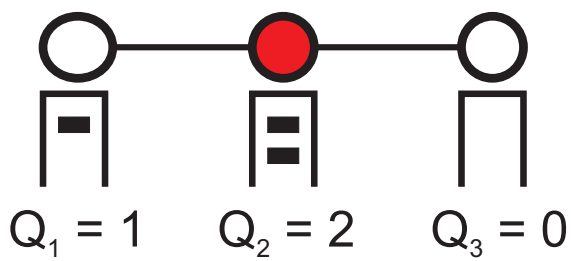
- linear network of 3 flows
- feasible schedules: $\Omega = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 3\}\}$
- MaxWeight scheduling compares $Q_2(t)$ and $Q_1(t) + Q_3(t)$



MaxWeight scheduling

For example:

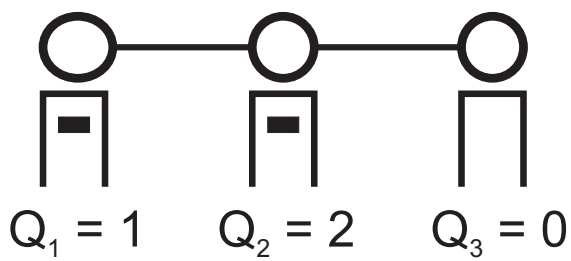
- linear network of 3 flows
- feasible schedules: $\Omega = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 3\}\}$
- MaxWeight scheduling compares $Q_2(t)$ and $Q_1(t) + Q_3(t)$

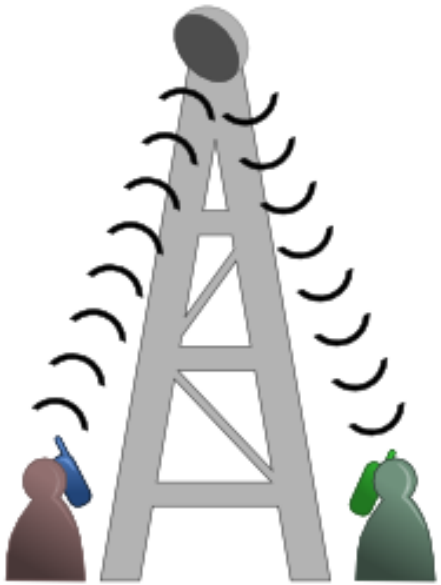


MaxWeight scheduling

For example:

- linear network of 3 flows
- feasible schedules: $\Omega = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 3\}\}$
- MaxWeight scheduling compares $Q_2(t)$ and $Q_1(t) + Q_3(t)$





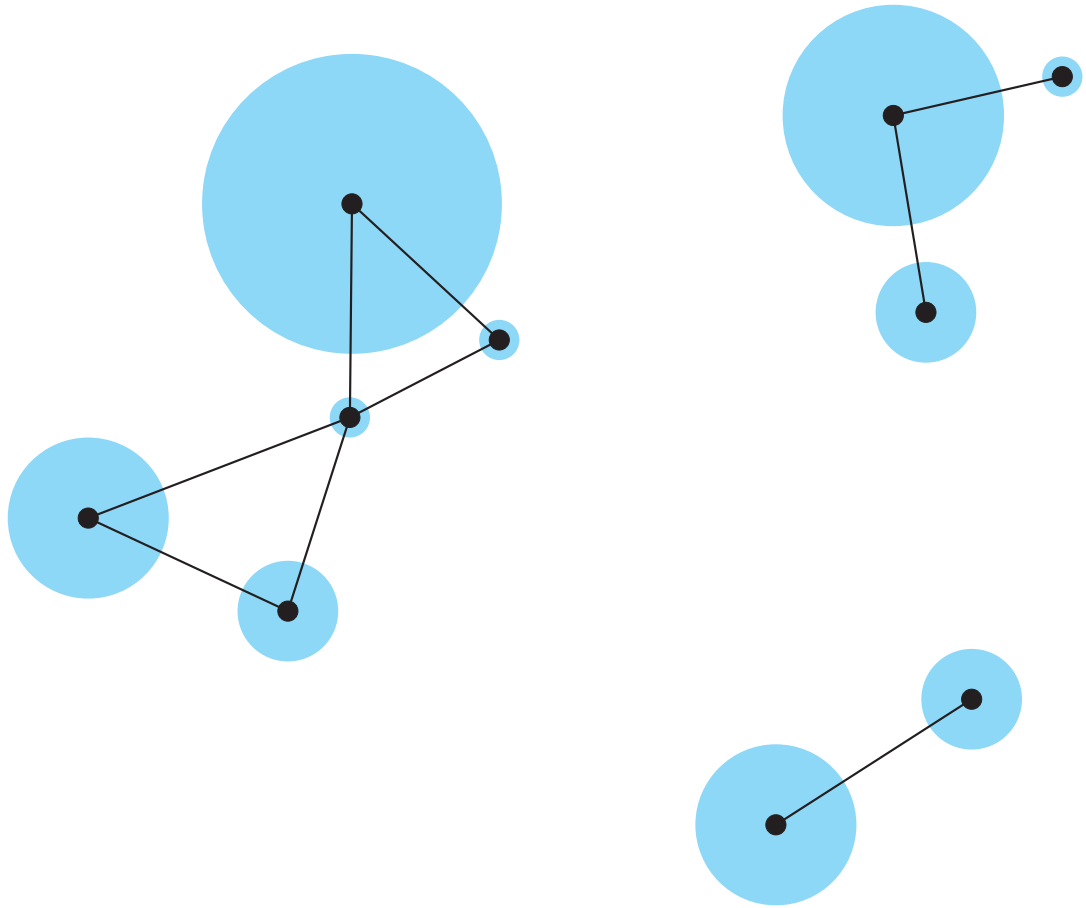
In reality, flows may be finite

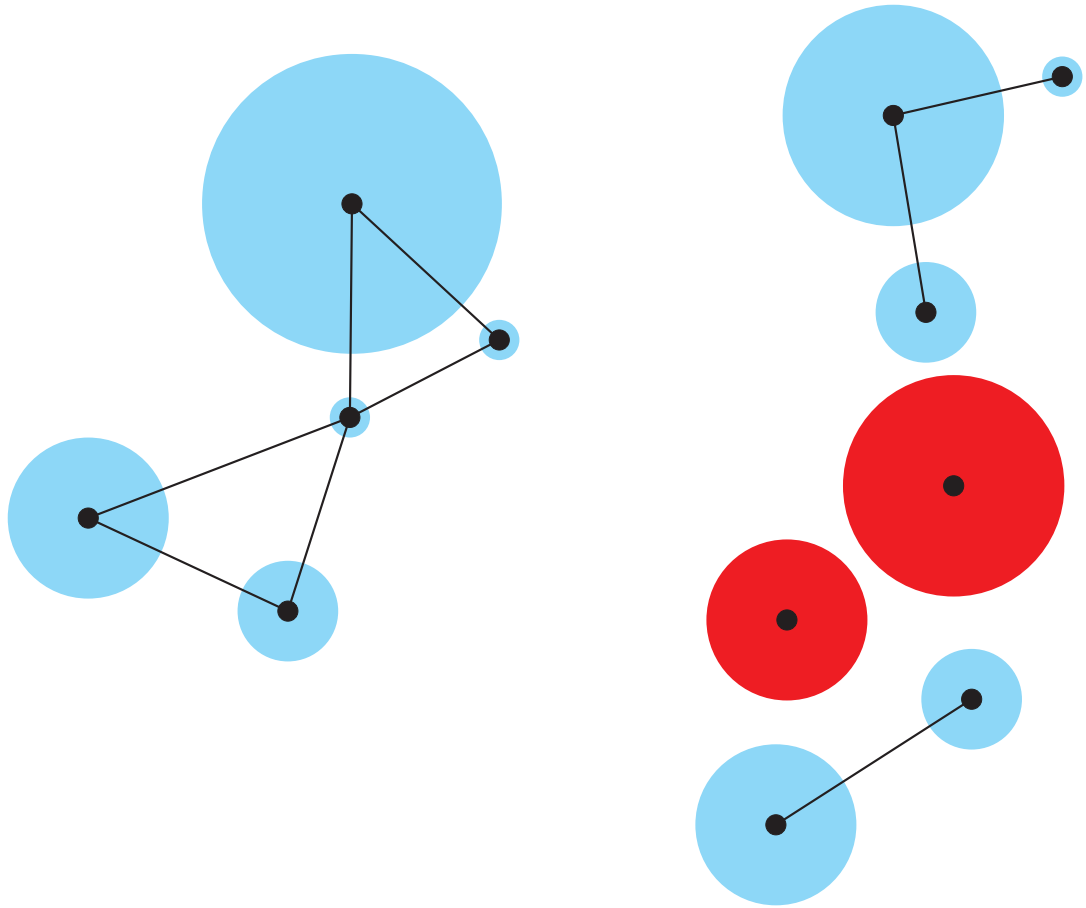
Flow-level dynamics

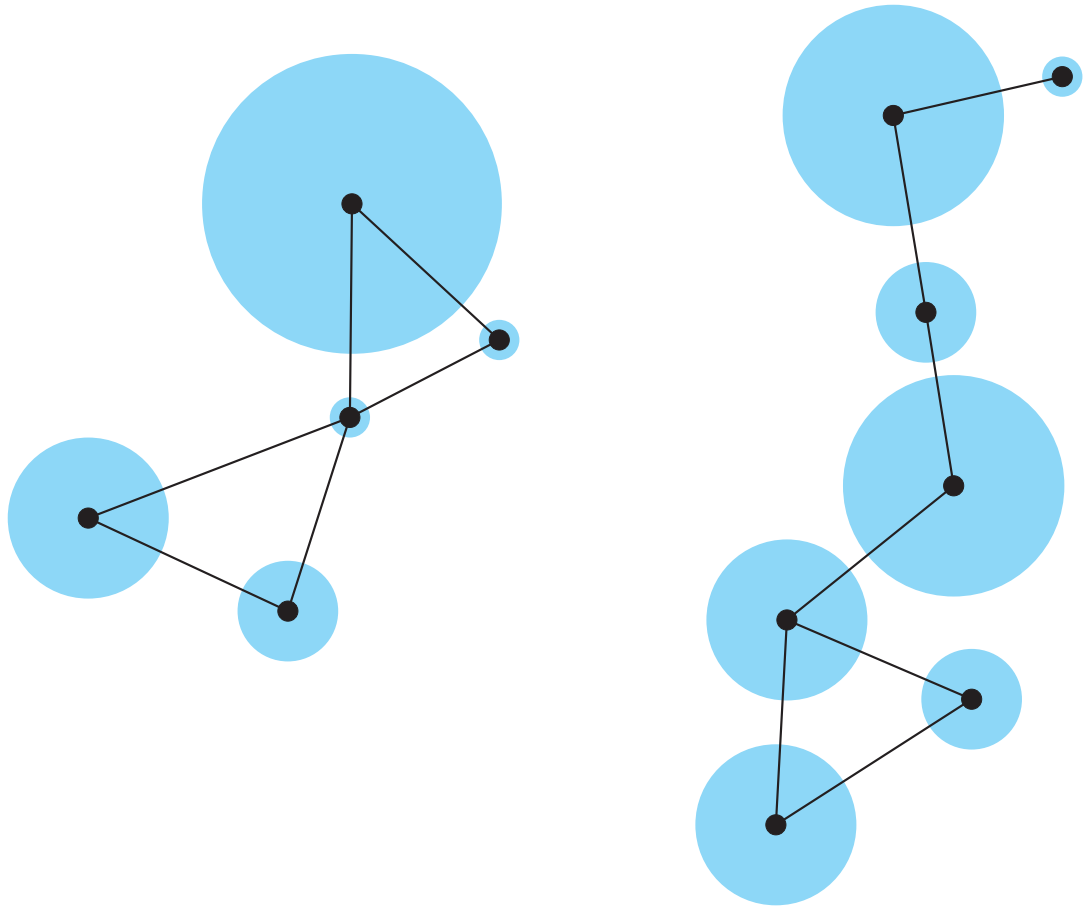
Flow-level dynamics

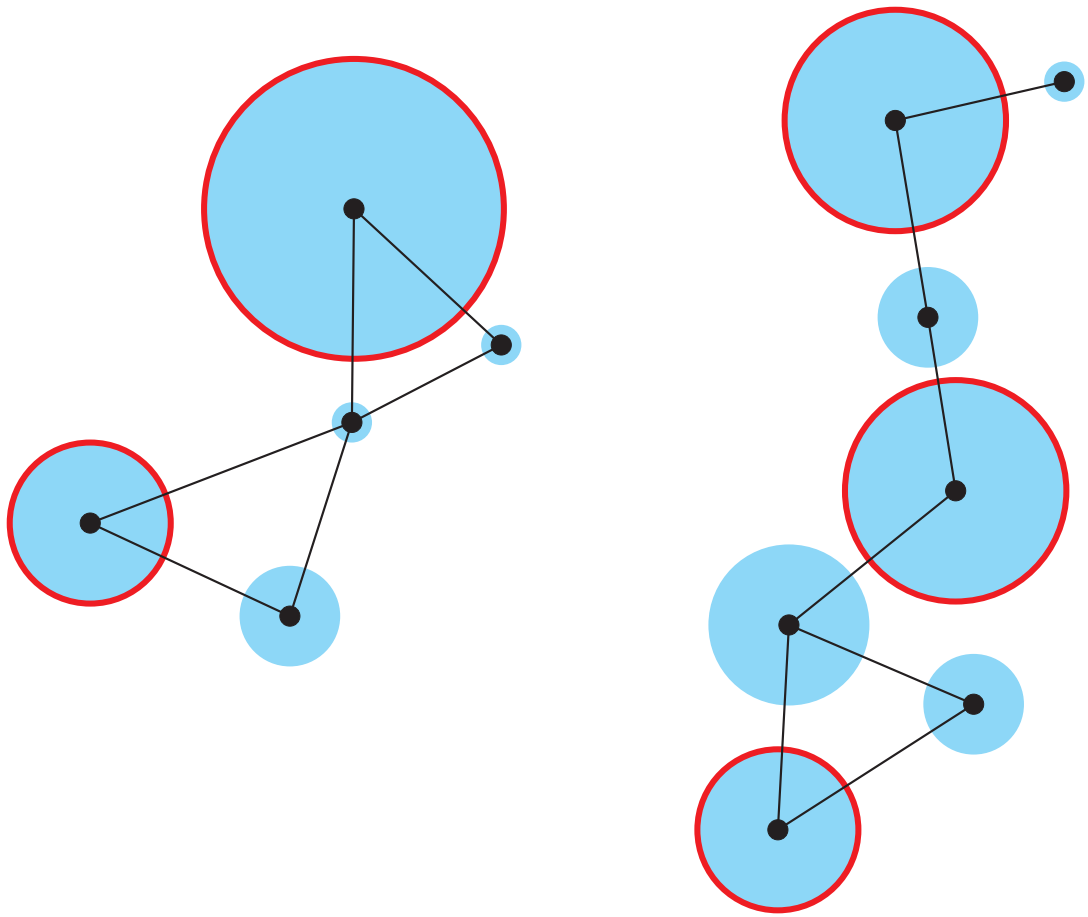
- new flows arrive occasionally
- flows only generate finite amount of traffic
- flows leave when completely served

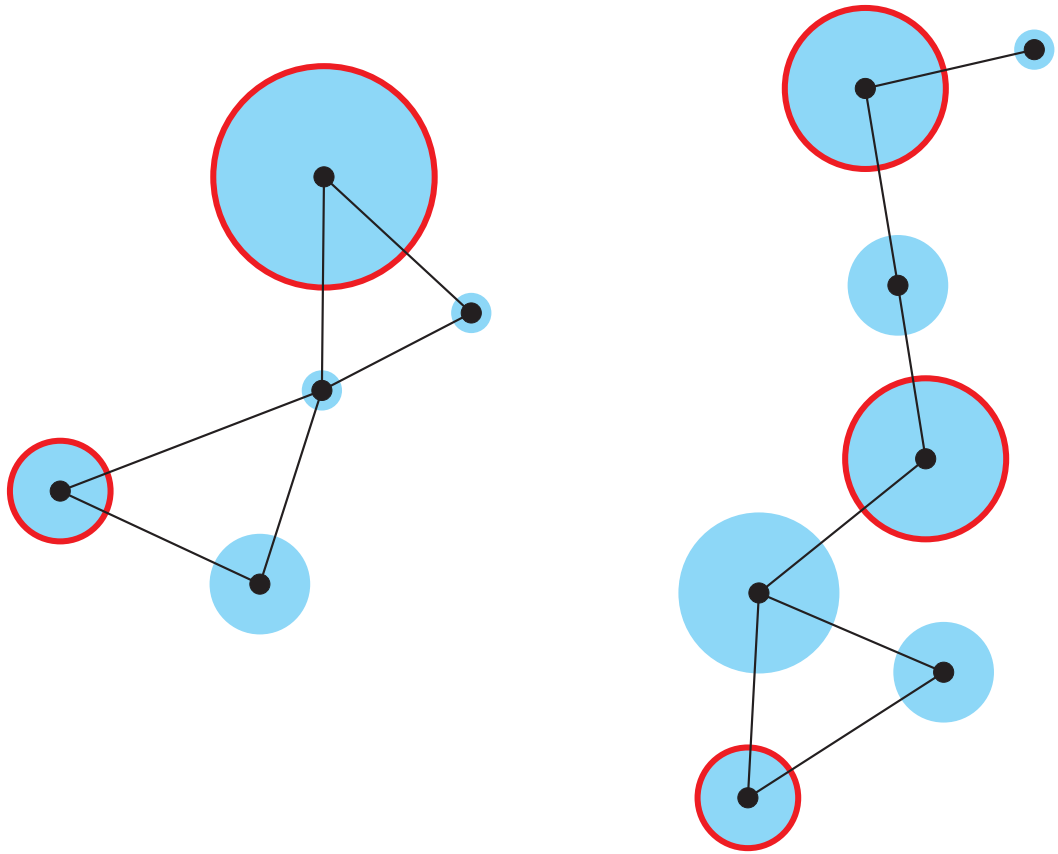
Flows have a certain location, and we assume that the interference constraints are based on distance

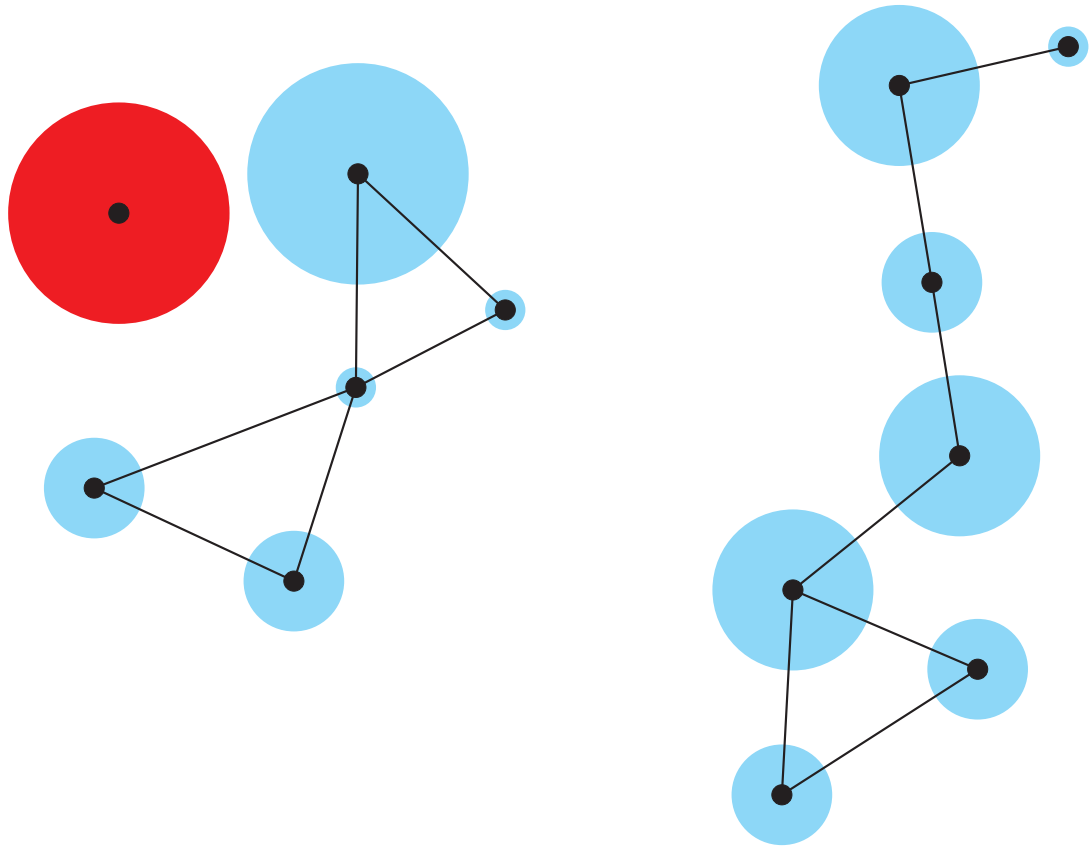


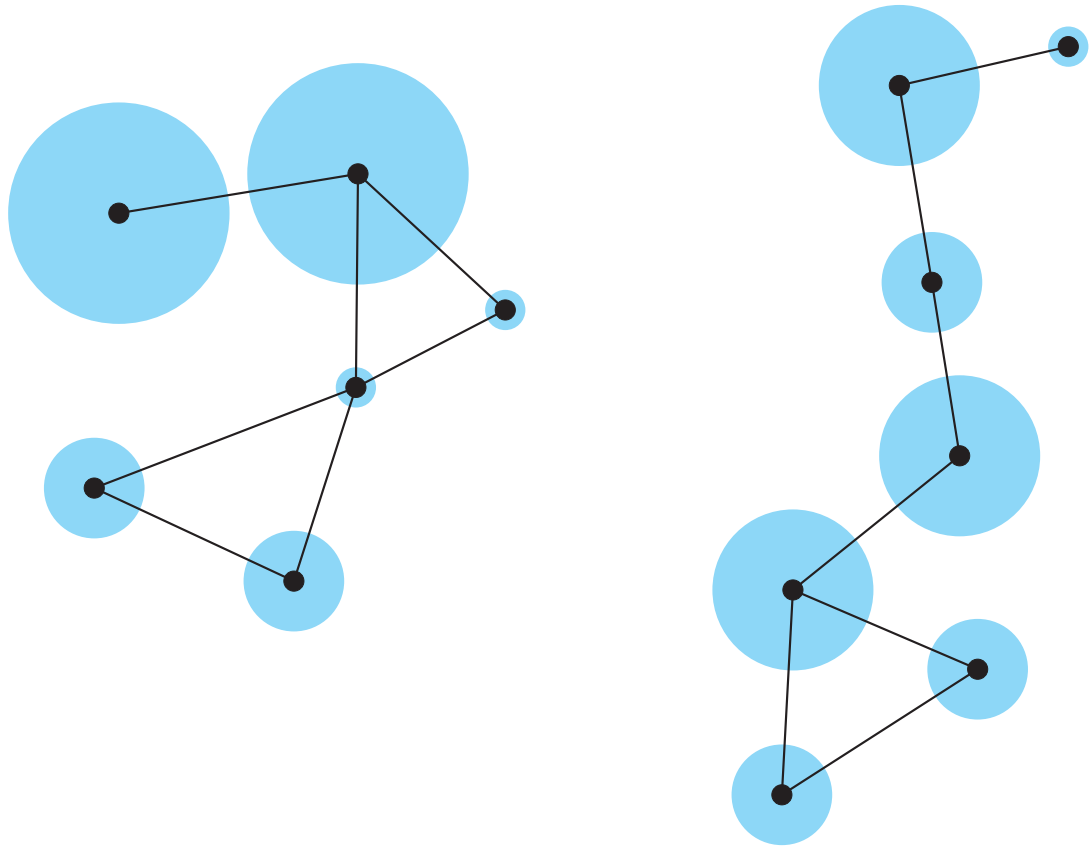


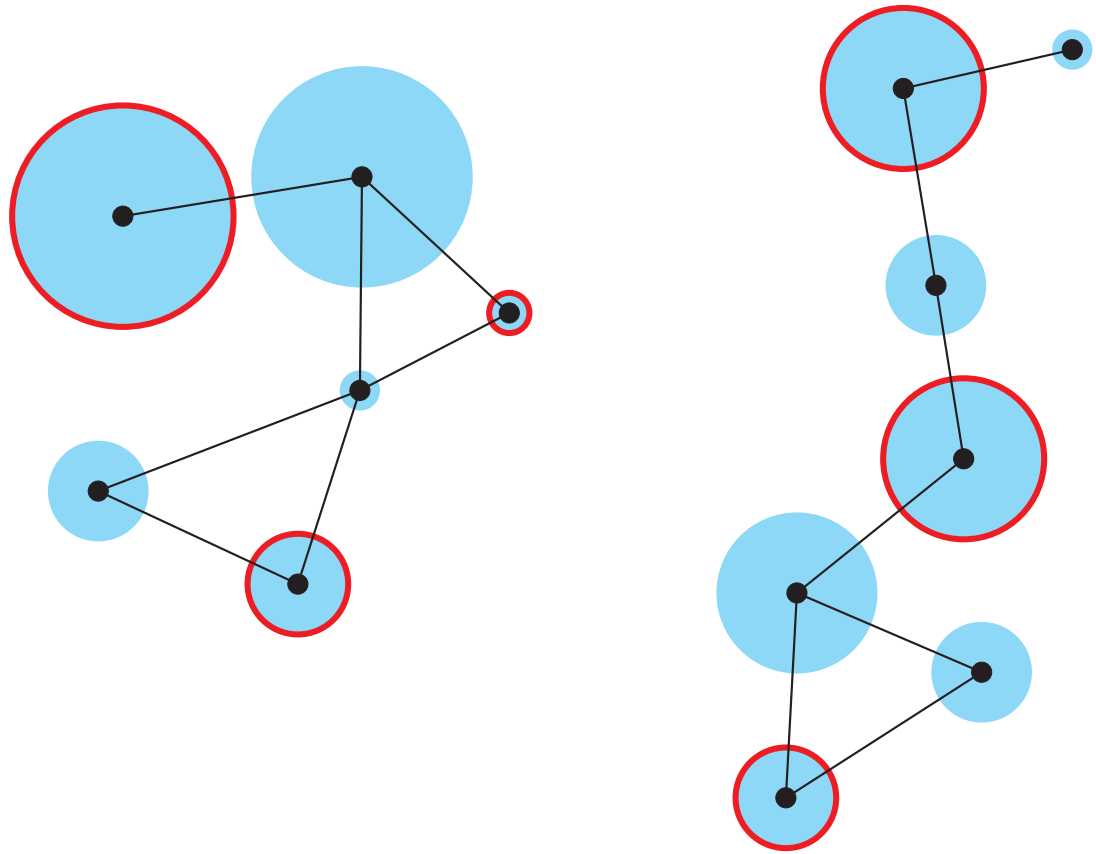


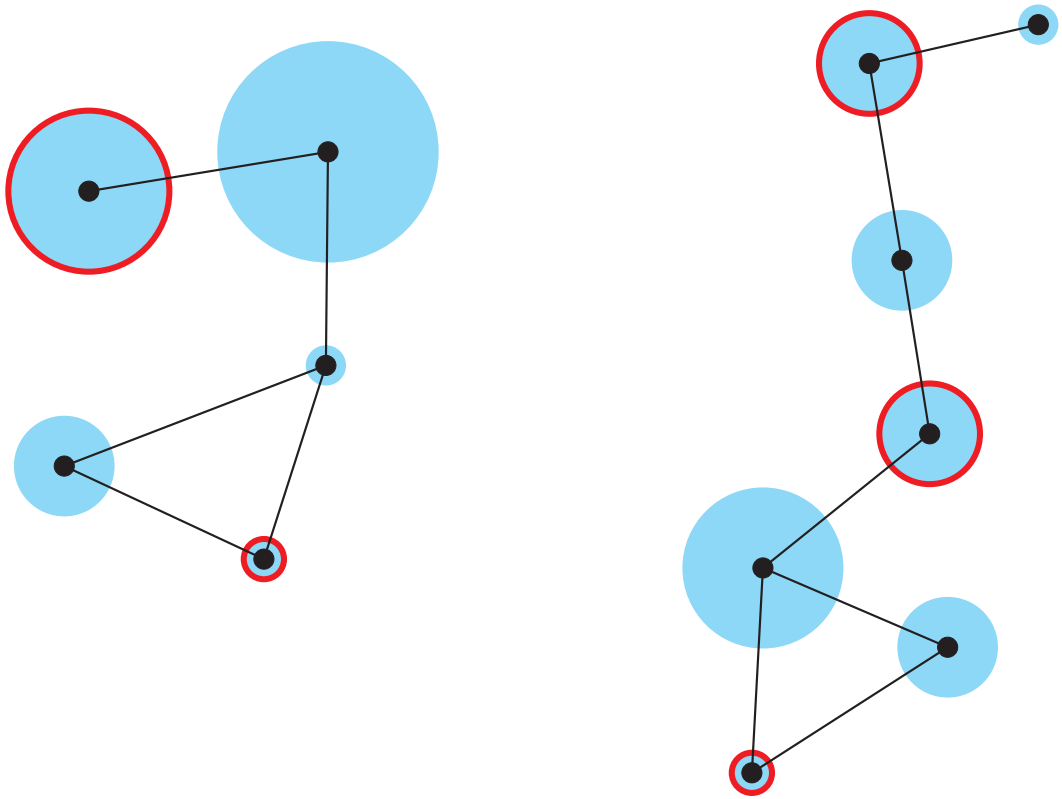






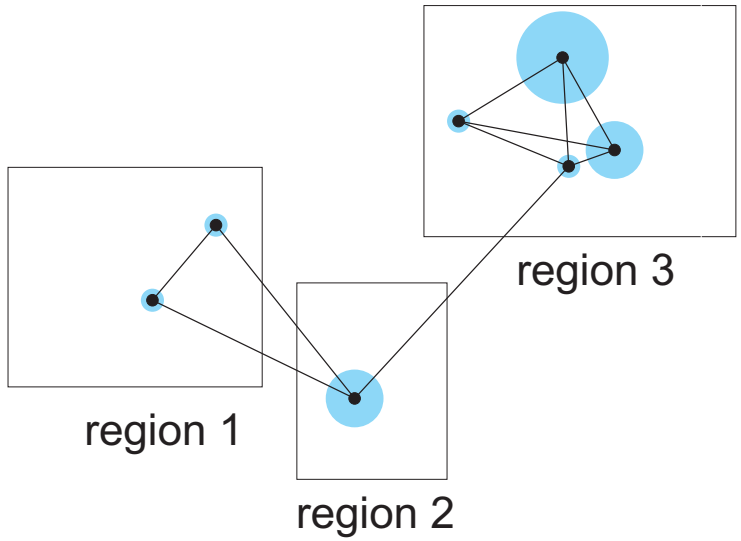






Instability examples

Example 1



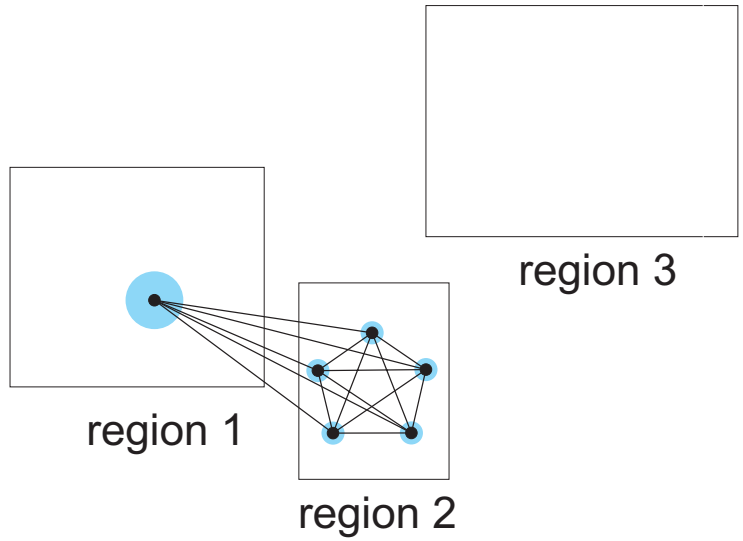
Denote

- λ_i arrival rate of flows in region i
- B_i the size of a new flow in region i
- $\rho_i = \lambda_i \mathbb{E}\{B_i\}$ the traffic load in region i

The capacity region of this network is given by

$$\rho_1 + \rho_2 < 1 \quad \text{and} \quad \rho_2 + \rho_3 < 1$$

Example 1

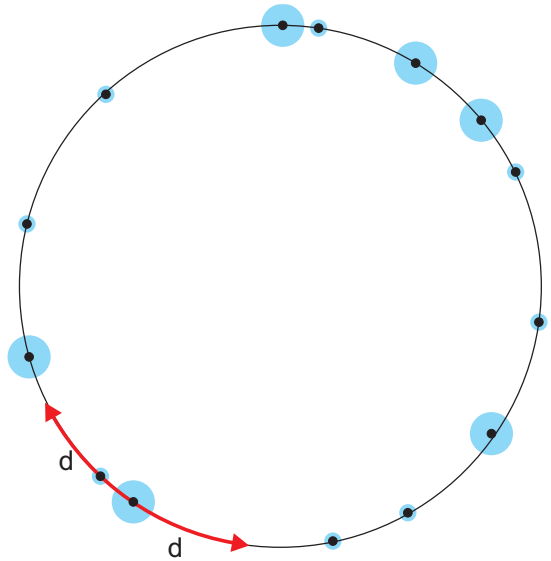


Now choose $B_2 \equiv 1$, then MaxWeight scheduling will always select $\{1, 3\}$ if at least one flow of size 2 or larger is present in region 1 or 3

The provides stability if

$$\rho_2 < (1 - \rho_1 + \lambda_1)(1 - \rho_3 + \lambda_3)$$
$$\rightarrow (1 - \rho_1)(1 - \rho_3)$$

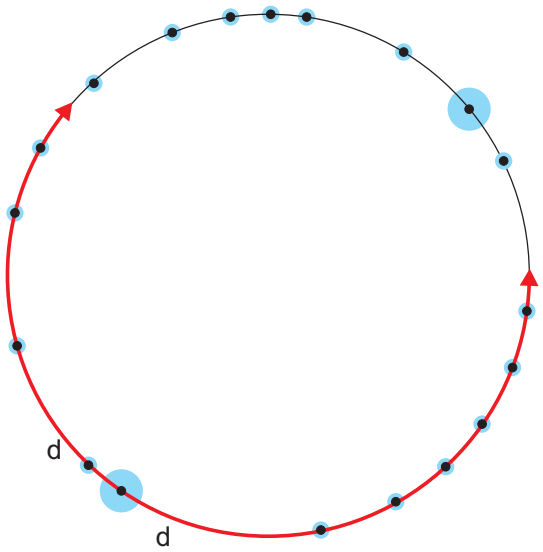
Example 2



- flows arrive uniformly on a ring with unit circumference
- new flows have size $B \equiv 2$
- scheduling is subject to reuse distance d

Example 2

Assume $d = 0.3$, so at most $M = 3$ flows can be scheduled simultaneously

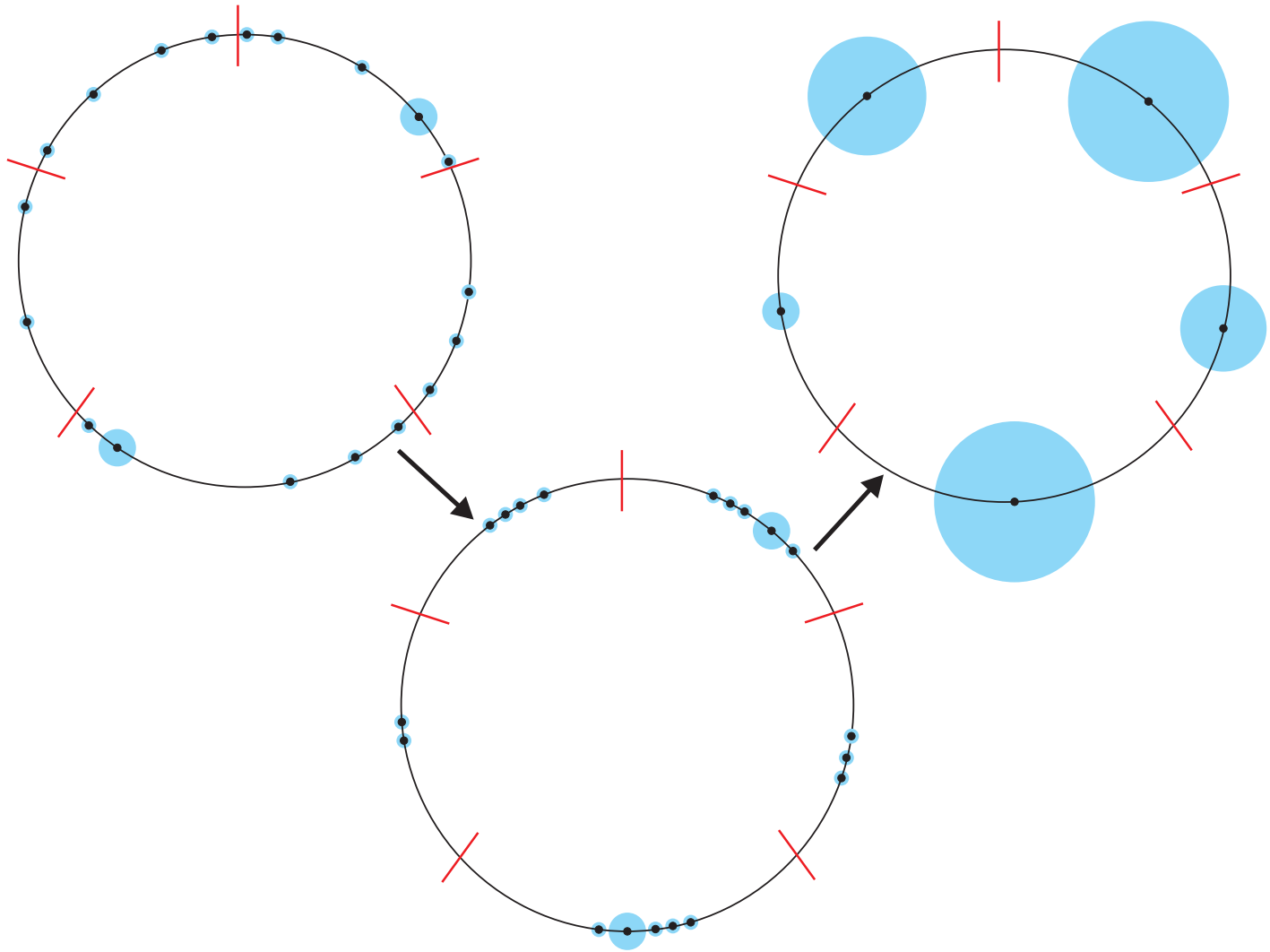


MaxWeight scheduling selects a set of flows with maximum aggregate backlog

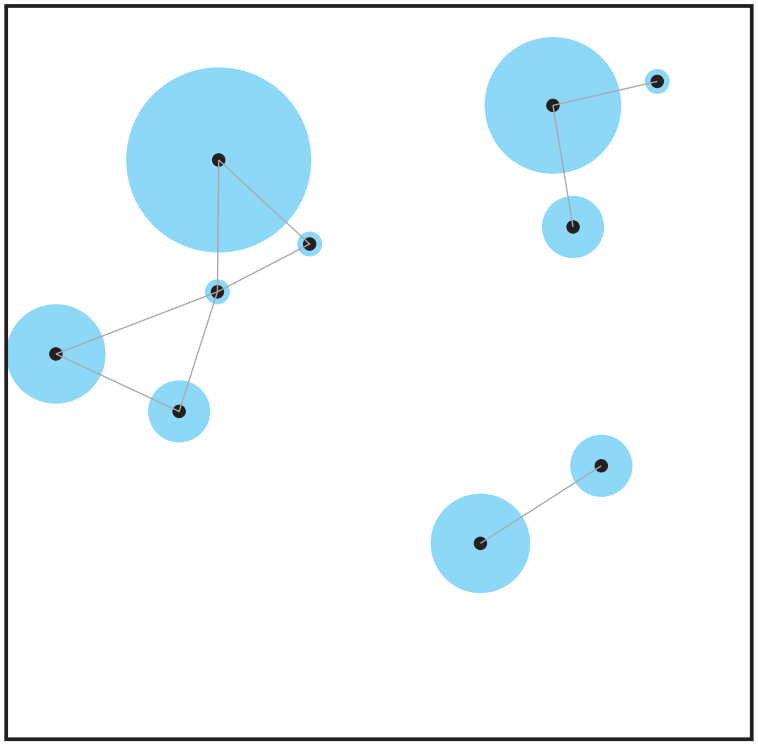
- flows of large size
- large number of flows

Put too much emphasis on flow size, ignores small flows

Region-based scheduling

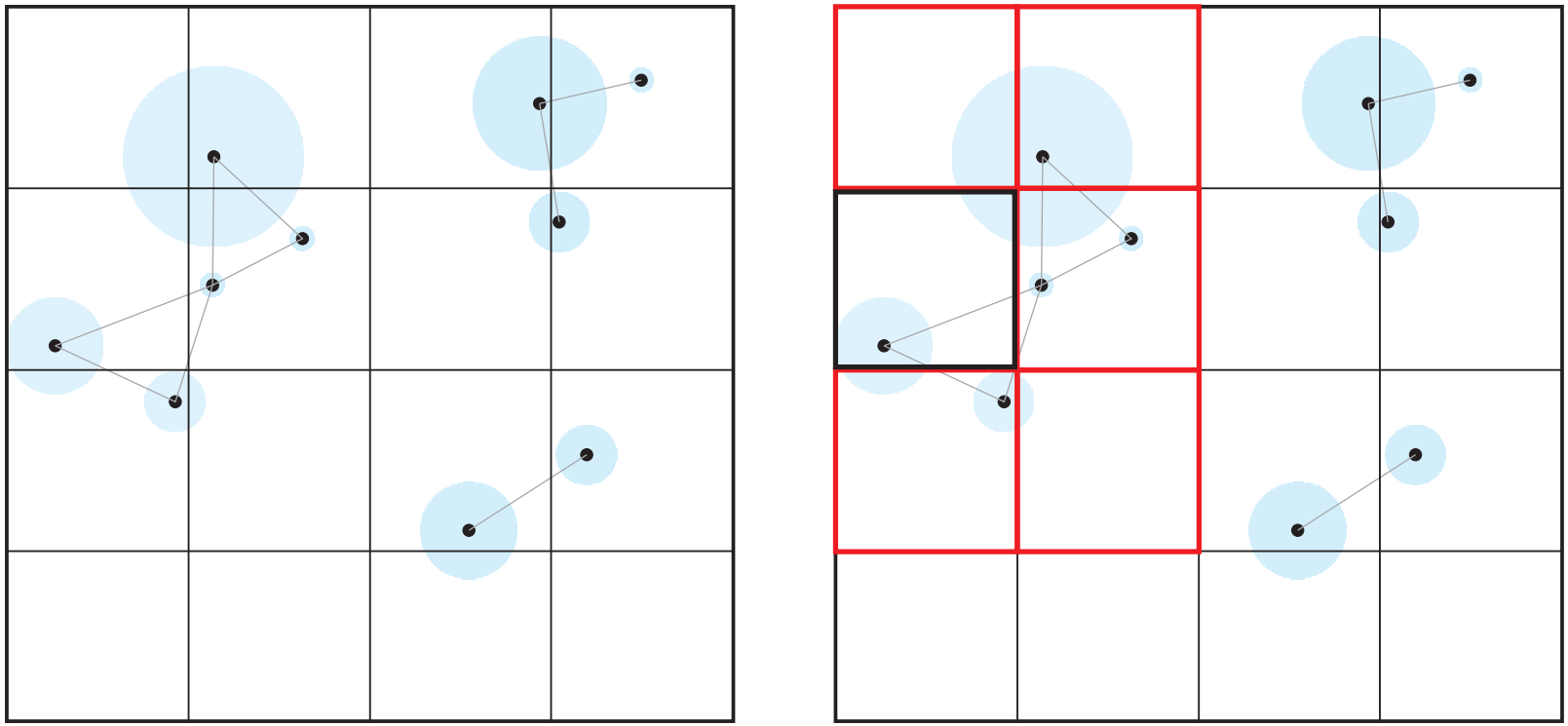


Solution: spatial aggregation



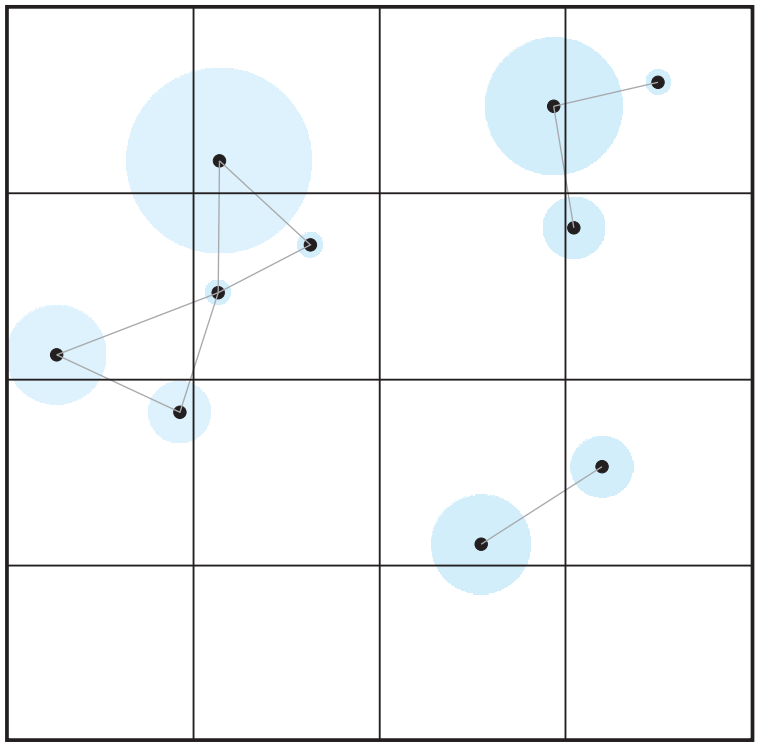
Consider

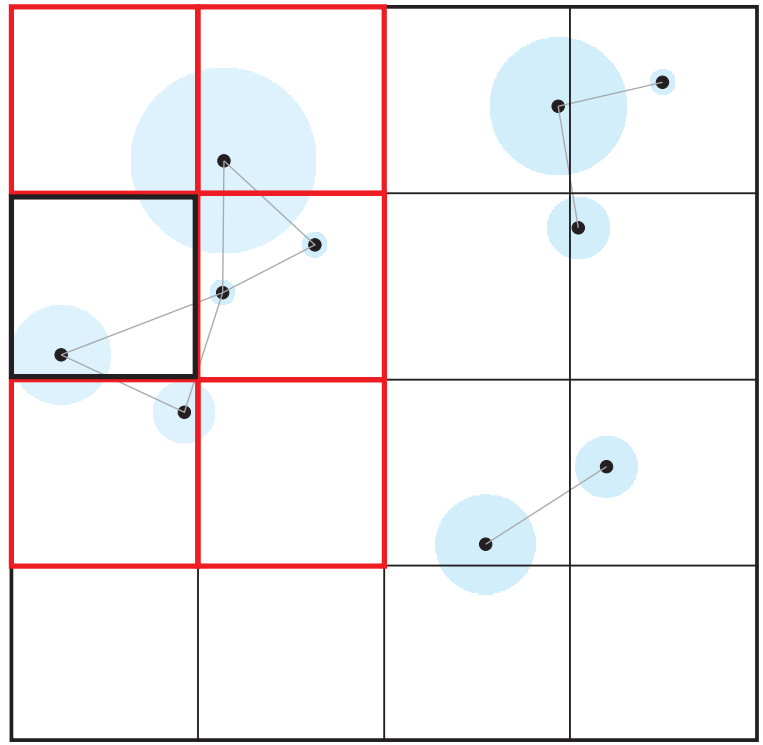
- a two-dimensional network on the unit square
- flows arrival at random locations
- new flows have random size
- scheduling constraints are based on reuse distance d



Partition the network into K^2 regions of size $1/K^2$

- schedule regions instead of flows
- compatible regions must be separated by at least distance d
- thus the scheduling constraints are more stringent





- the capacity region $C(K, d)$ under K-partition is smaller than the full capacity region $C(d)$
- but, for K-partition we know a throughput-optimal scheduler
  $\rightarrow$ (region-based) MaxWeight scheduling!

How do $\mathcal{C}(K, d)$ and $\mathcal{C}(d)$ compare?

For all $K \geq 1$, we have that $\mathcal{C}(K, d) \subseteq \mathcal{C}(d)$. These 'converge' as K grows large:

How do $\mathcal{C}(K, d)$ and $\mathcal{C}(d)$ compare?

For all $K \geq 1$, we have that $\mathcal{C}(K, d) \subseteq \mathcal{C}(d)$. These 'converge' as K grows large:

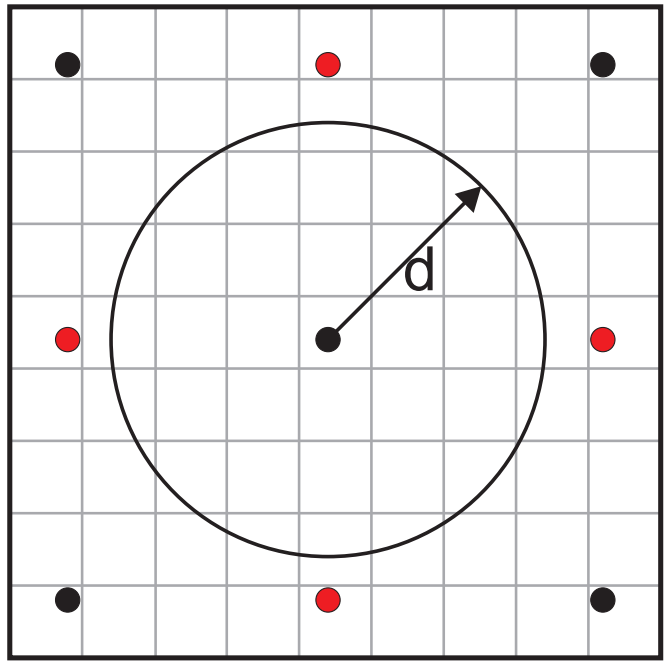
Theorem

Let λ be an arrival density function such that $(1 + \epsilon)\lambda \in \mathcal{C}(d)$ for some $\epsilon > 0$. Then there exists a $K = K(\lambda)$ such that $\lambda \in \mathcal{C}(K, d)$.

However, $\mathcal{C}(K, d)$ and $\mathcal{C}(d)$ will in general not coincide for any value of K

Proposition

Let $K \geq 1$, then there exists an arrival density function $\hat{\lambda}$ such that $\hat{\lambda} \in \mathcal{C}(d)$, but $\left(\frac{1}{2} + \epsilon\right) \hat{\lambda} \notin \mathcal{C}(K, d)$ for any $\epsilon > 0$.



The 9-partition with reuse distance $d = 0.35$

Conclusions and outlook

We studied MaxWeight scheduling under flow-level dynamics

- MaxWeight scheduling is no longer throughput-optimal
- we propose region-based scheduling that partitions the network into a finite number of regions
- region-based MaxWeight scheduling provides stability

Conclusions and outlook

We studied MaxWeight scheduling under flow-level dynamics

- MaxWeight scheduling is no longer throughput-optimal
- we propose region-based scheduling that partitions the network into a finite number of regions
- region-based MaxWeight scheduling provides stability

But:

- how to choose the regions (tradeoff complexity - capacity)?
- given a class of arrival densities, can we find a stabilizing partitioning?

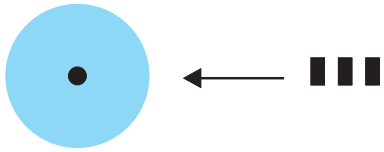
Transmission rate variations

So far we assumed fixed transmission rates $R_i(t) \equiv 1$

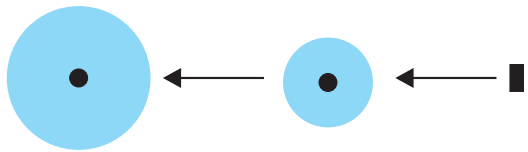
Rate variations cause instability of MaxWeight for single-channel case (VdV, Borst and Shneer (2009)). Solutions:

- Sadiq and de Veciana (2009): delay-based MaxWeight scheduling
- Liu, Ying and Srikant (2010,2011): schedule according to (maximum) transmission rate

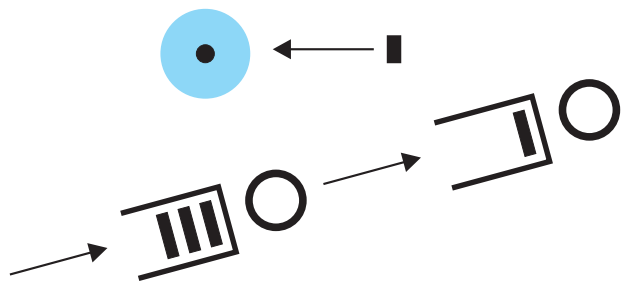
Can we adapt devise an algorithm to handle both rate variations and spatial inefficiency?



gradual traffic



multi-hop



combining persistent & finite flows