Dynamic Monopolies, Cellular Automata, and Network Decontamination

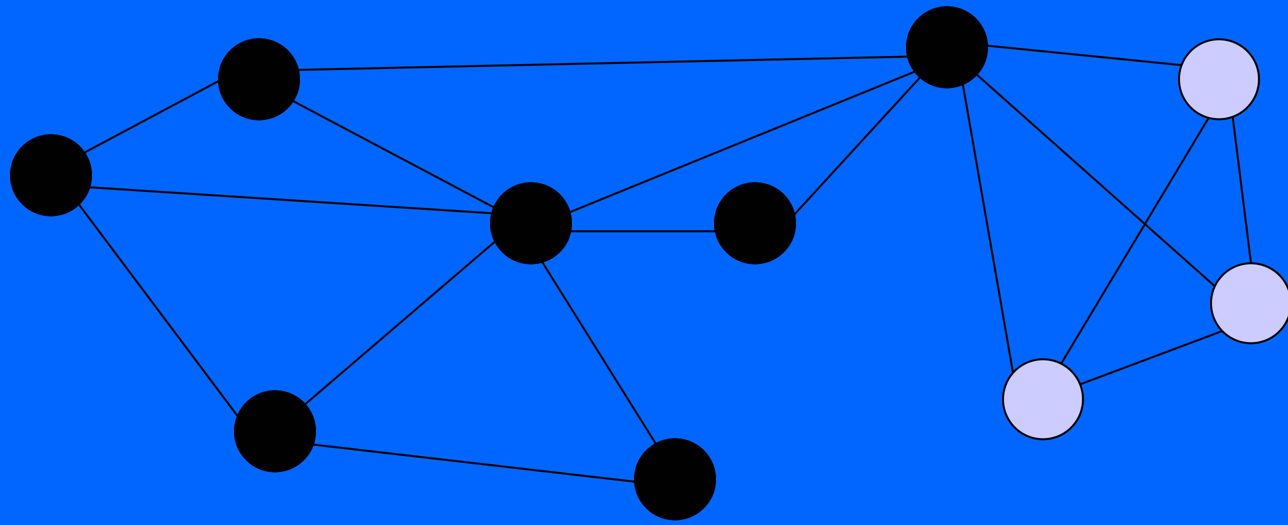
P. Flocchini
University of Ottawa

The Problem

Graph G=(V,E)

A node is either black or white.

A node (a)synchronously re-colors itself with the color held by the majority of its neighbors.



Majority Voting - Literature

In Dynamical Systems

[Goles et al 80,85,…]

[Agur et al 88]

[Granville 91]

[Moran 94,95 …]

and more …

Study of periodic behaviors, transients, number of fixed points in various graph structures …

Majority Voting - Literature

In Distributed Computing

Catastrophic faults [Santoro et al 90,94,95] ...

Monopolies [Bermond et al. 95,96] [Peleg 96] ...

Dynamic Monopolies

[Peleg 97]

Tori: [Flocchini et al. 98,04]

Chordal Rings: [Flocchini et al. 98,01]

Butterfly: [Luccio et al. 99]

Trees: [Kralovic 01]

Interconnection Networks: [Flocchini et al. 03]

Motivation: Propagation of faults in majority-based distributed systems

Local Majority is employed at each site
- reliability
- fault-tolerance

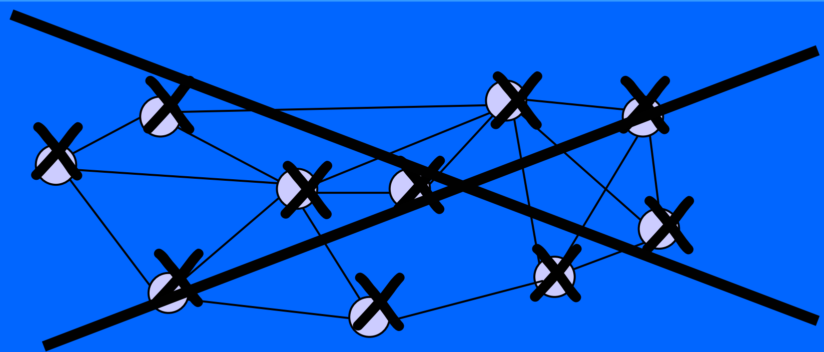
e.g.
- consensus and agreement protocols
- data consistency protocols in quorum systems
- key distribution in security
- reconfiguration in system level analysis

Motivation: Propagation of faults in majority-based distributed systems

Graph = distributed system

Black node = (permanent) faulty node

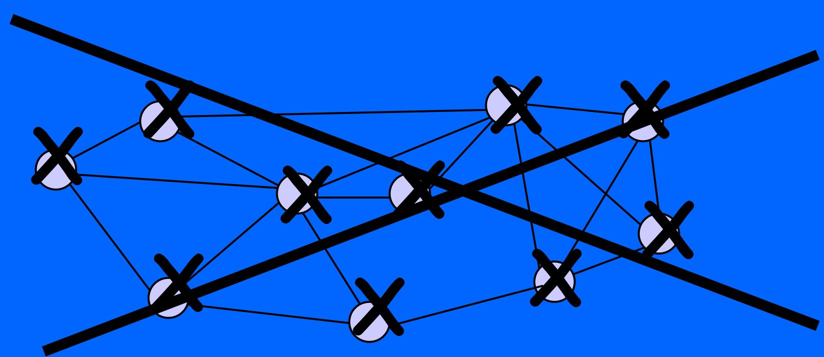
A node becomes faulty when the majority of its neighbors are faulty



Dynamo = pattern of initial faults which leads the entire system to a faulty behavior

Motivation: Propagation of faults in majority-based distributed systems

MONOCHROMATIC FIXED POINT

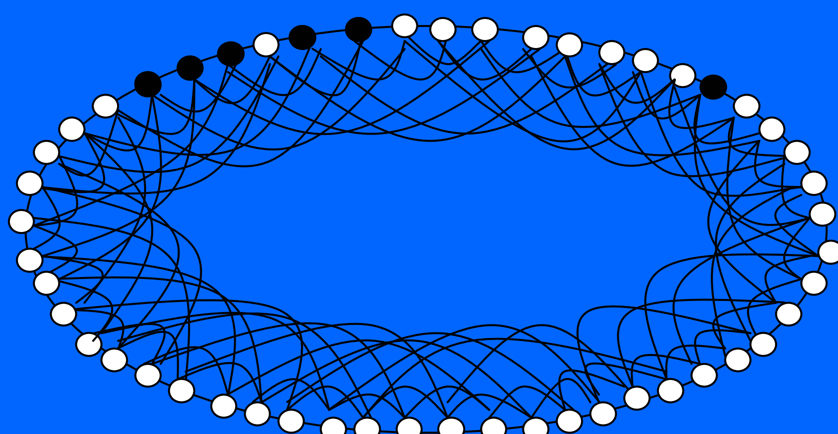


Dynamo = pattern of initial faults which leads the entire system to a faulty behavior

Interesting Questions in this context:

What are the patterns of faults which eventually corrupt the entire system ?

What is the minimum number of faulty nodes that eventually corrupt the entire system ?



Different models

[Peleg 96]

Asynchronous / Synchronous

Self-including / self-not-including

In case of tie:

simple/strong majority
prefer-black
prefer white
prefer change
prefer myself

Reversible/ Irreversible / Monotone

Irreversible and Monotone Dynamos

Irreversible: An initially black node never changes (permanent faults)

Monotone: At any time the set of black nodes must include the set of the previous step.

$B(t+1) \supseteq B(t)$

$B(t)$ = set of black nodes at time t

In this case asynchronous and synchronous dynamics are equivalent

Observation

The reversible Ring is a circular 1-dim CA

Reversible (prefer-black)

Reversible (prefer-self)

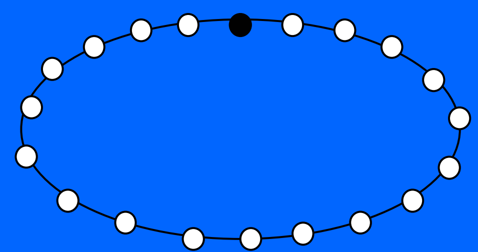
000	0	0
001	1	0
010	0	0
011	1	1
100	1	0
101	1	1
110	1	1
111	1	1
	250	232

Example: Simple Majority

Observation

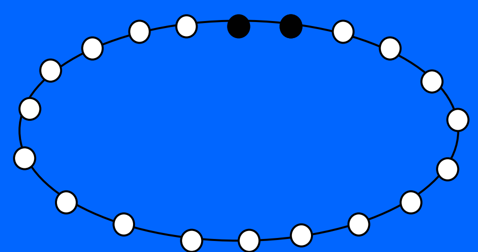
The reversible Ring is a circular 1-dim CA

irreversible model
(self/non-self)



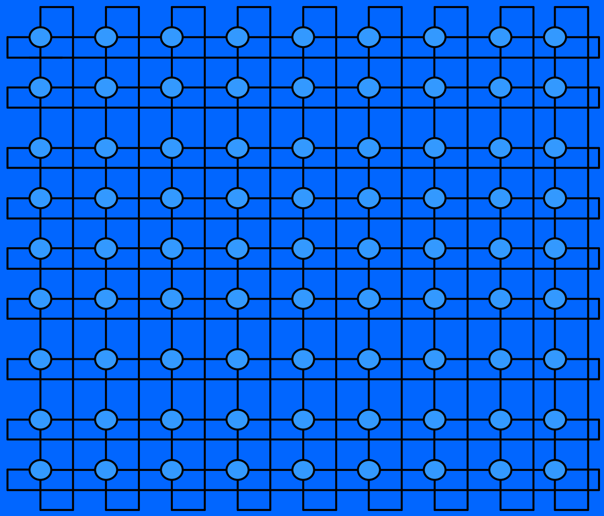
Example: Simple Majority
Optimal Dynamos

reversible model
(self/non-self)



IN TORI

(2-dim CAs)



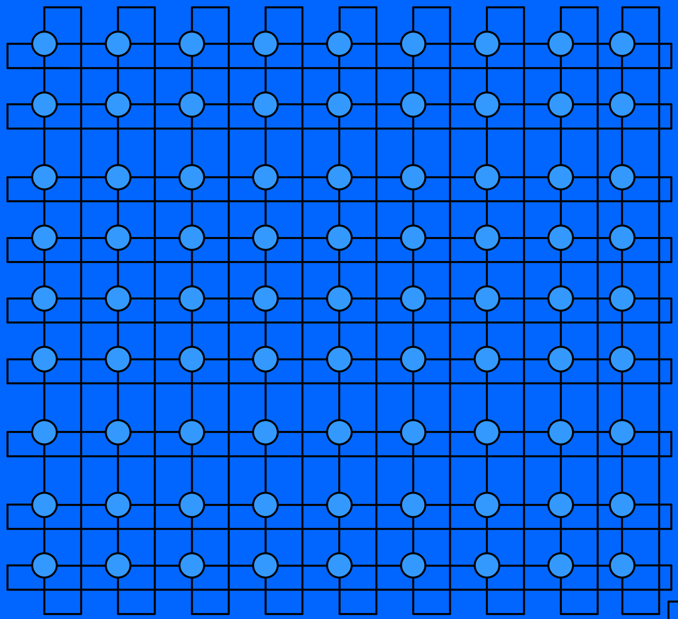
Upper and Lower bounds on size of a dynamo (Irreversible/Monotone) for:

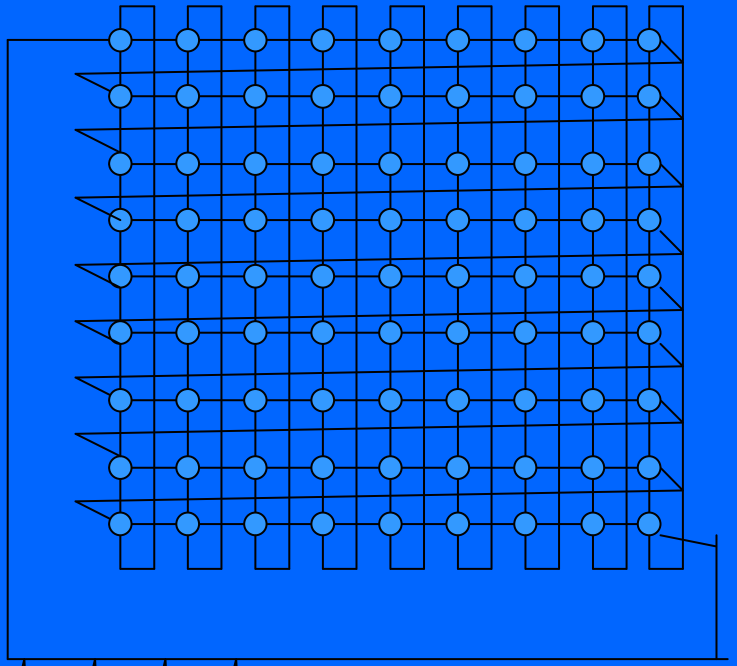
Various Toroidal Structures

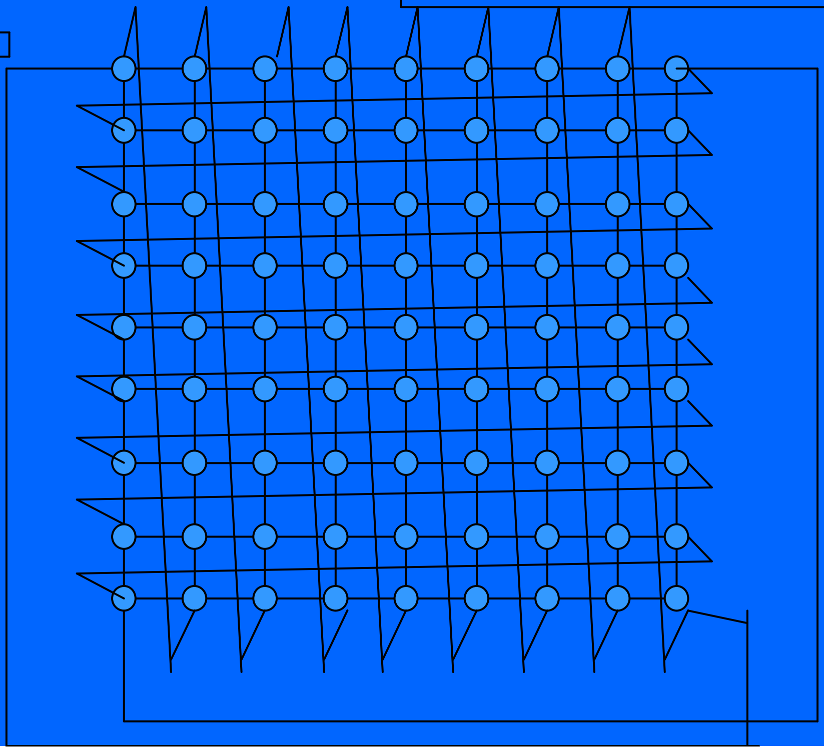
Simple/Strong Majority

[Theoretical Computer Science: Flocchini et al. 98,04]

Toroidal Meshes







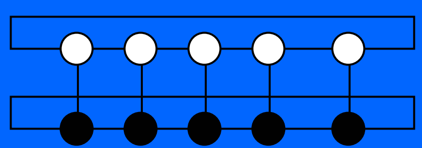
Irreversible Dynamos				
	Simple Majority		Strong Majority	
	Lower Bound	*Upper Bound*	*Lower Bound*	*Upper Bound*
Toroidal mesh	$\lceil \frac{m+n}{2} \rceil - 1$	$\lceil \frac{m+n}{2} \rceil - 1$	$\lceil \frac{mn+1}{3} \rceil$	$\lceil \frac{H}{3} \rceil (K+1)$
Torus cordalis	$\lceil \frac{n}{2} \rceil$	$\lfloor \frac{n}{2} \rfloor + 1$	$\lceil \frac{mn+1}{3} \rceil$	$\lceil \frac{m}{3} \rceil (n+1)$
Torus serpentinus	$\lceil \frac{N}{2} \rceil$	$\lfloor \frac{N}{2} \rfloor + 1$	$\lceil \frac{mn+1}{3} \rceil$	$\lceil \frac{H}{3} \rceil (K+1)$

Monotone Dynamos				
	Simple Majority		Strong Majority	
	Lower Bound	*Upper Bound*	*Lower Bound*	*Upper Bound*
Toroidal mesh	$m+n-2$	$m+n-1$	$\lceil \frac{mn+1}{2} \rceil$	$\lceil \frac{mn}{2} + \frac{N}{6} + \frac{2}{3} \rceil^*$
Torus cordalis	$n+1$	$n+1$	$\lceil \frac{mn+1}{2} \rceil$	$\lceil \frac{mn}{2} + \frac{N}{6} + \frac{2}{3} \rceil^*$
Torus serpentinus	$N+1$	$N+1$	$\lceil \frac{mn+1}{2} \rceil$	$\lceil \frac{mn}{2} + \frac{N}{6} + \frac{2}{3} \rceil^*$

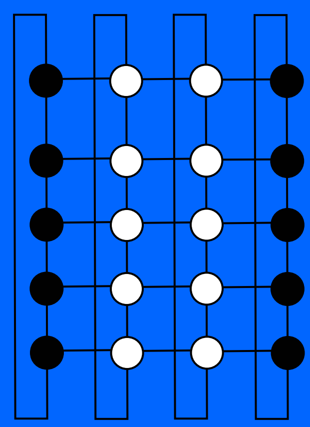
Table 1: Bounds on the size of monotone dynamos, for tori of $m \times n$ vertices; $N = min\{m, n\}$, and $H, K = m, n$ or $H, K = n, m$ (choose the alternative that yields stricter bounds). The asterisk denotes a worst case.

White Blocks

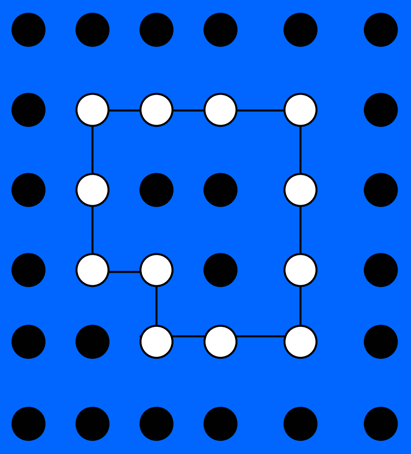
White patterns that CANNOT appear in a dynamo



strong white block for
toroidal mesh



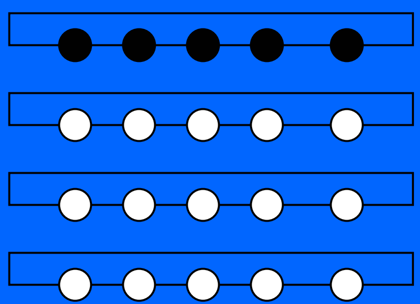
- simple white block for toroidal mesh & cordalis
- strong for serpentinus



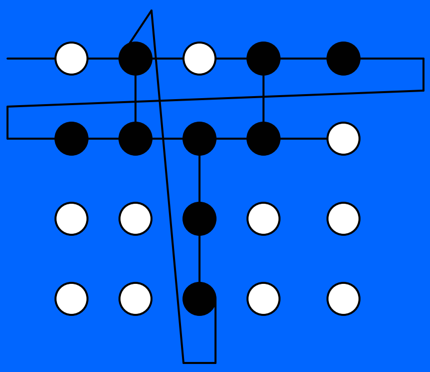
strong white block
in any tori

Black Compacts

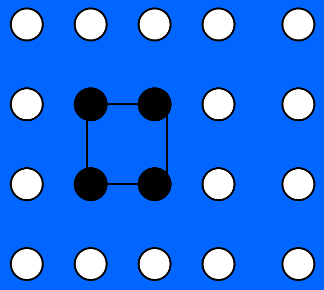
Black patterns that MUST appear in a dynamo



In a toroidal mesh



In a torus cordalis



In any tori

In the case of Monotone Dynamos

T = nodes of the torus
S = nodes of a simple (strong) monotone dynamo

S is a monotone dynamo under the simple(strong) majority rule iff :

S is a collection of black compacts

guarantees monotonicity

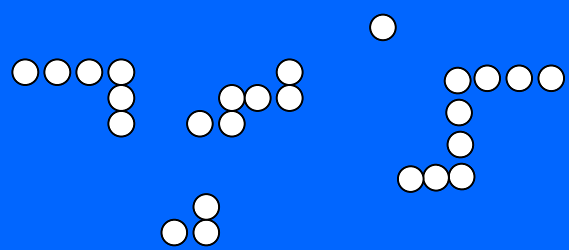
T- S does not contain any simple (strong) white block

guarantees convergence

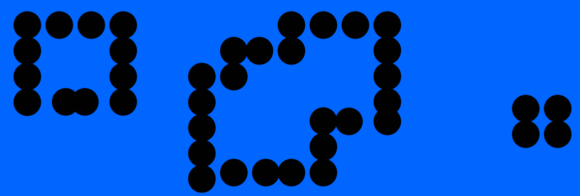
All Tori - Strong Majority- Monotone:

Lower Bound

Subset of initial white must be a forest



Black compacts = cycles



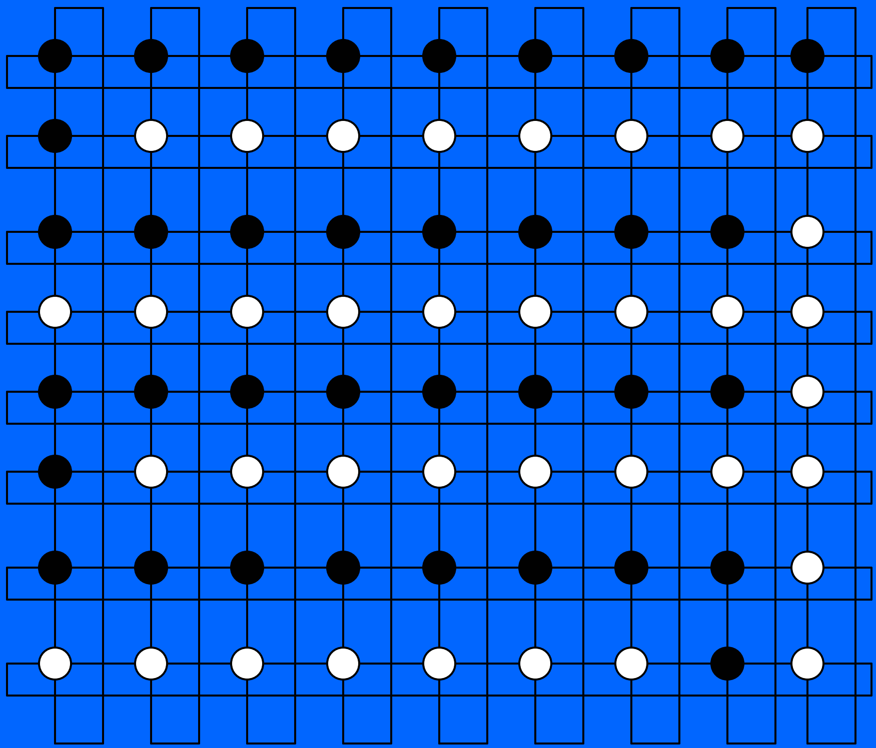
A monotone dynamo S for a torus m x n has size $\geq \lceil (mn+1)/2 \rceil$

All Tori: Upper Bound

m and/or n even

In any m x n toroidal mesh there exists a monotone dynamo of size (mn)/2 + 1

N = min{m,n}
M = max{m,n}

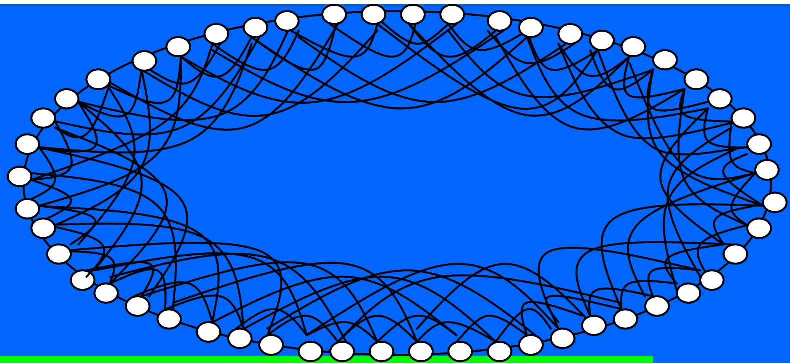


Evolution time:

N + M/2 - 2 (m, n even)

n + m/2 - 2 (m even n odd)

steps.



CHORDAL RINGS

Optimal Dynamos and characterization of the optimal patterns

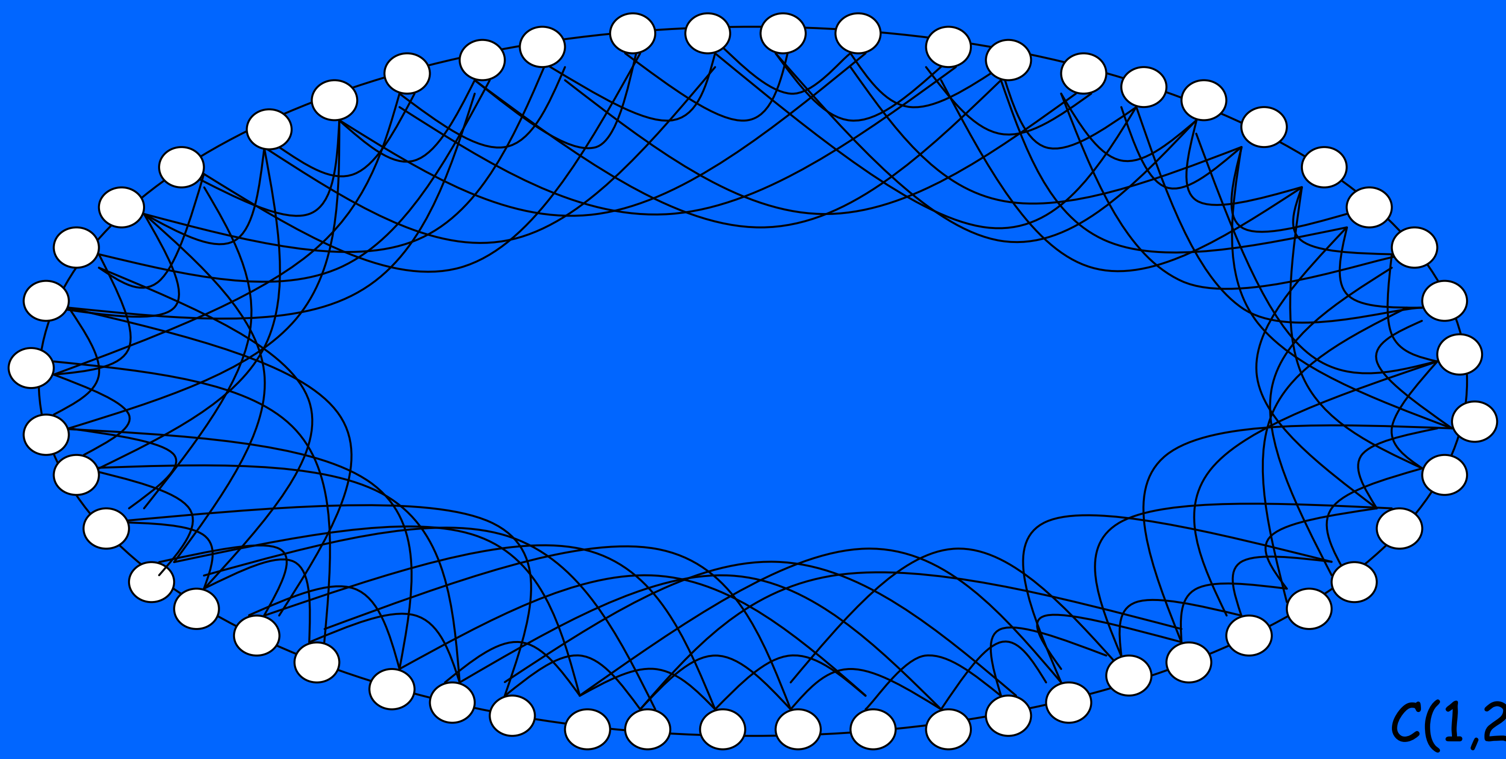
- Irreversible
- Simple Majority (= self-not, prefer black)

[Discrete Applied Mathematica: Flocchini et al. 98,01]

Chordal Rings

$C(1, d_2, \ldots, d_h)$

[particular cases: double loop $C(1,k)$
triple loop $C(1, 2, k)$]



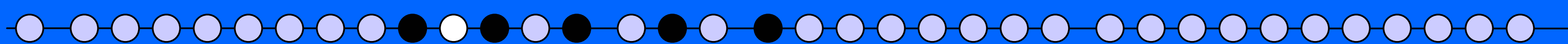
$C(1,2,7)$

Double Loop C<1,k>

An optimal dynamo has size (k+1)/2

k odd: only one pattern

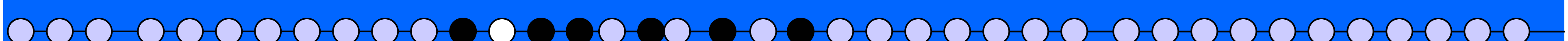
$(\bullet\ \circ)^{k/2}(\bullet)$



Ex. for C<1,9>

k even: optimal dynamo of the form

$(\bullet\ \circ)^{a}(\bullet)\ (\bullet\ \circ)^{b}\ (\bullet)$ with $a+b= (k-2)/2$



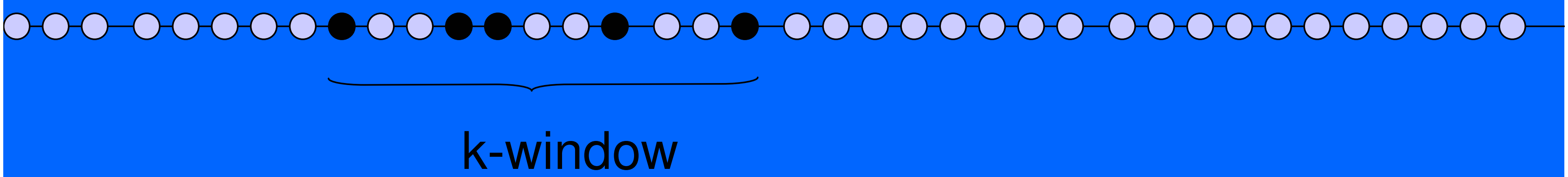
Ex. for C <1,10>

Triple Loop C<1,2,k>

An optimal dynamo has size (k+4)/3

A configuration of size k is an optimal dynamo iff:

- it contains two consecutive black nodes
- it does not contain three consecutive white nodes



Ex. C<1,2,11>

All and Only Optimal Dynamos for Triple Loops

k=3i+2

i+1 $(\bullet\circ\circ)^a(\bullet\bullet)\ (\circ\circ\bullet)^b$ a+b=i

k=3i+1

i(i+1)

$(\bullet\circ\circ)^a(\bullet\circ)\ (\bullet\circ\circ)^b(\bullet\bullet)(\circ\circ\bullet)^c$ a+b+c =i-1

$(\bullet\circ\circ)^a(\bullet\bullet)\ (\circ\circ\bullet)^b(\circ\bullet)(\circ\circ\bullet)^c$

k=3i

$(i^3+i^2)/2$

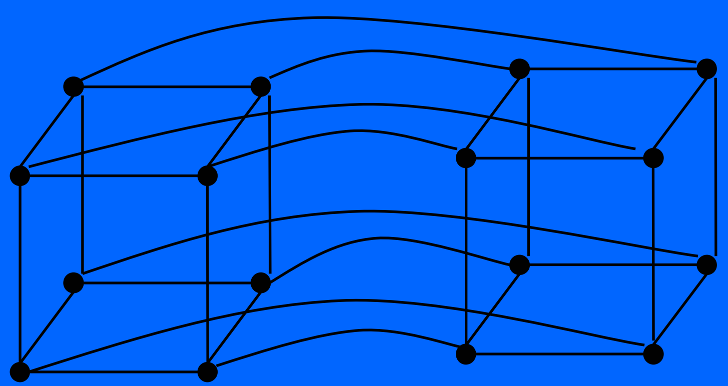
$(\bullet\circ\circ)^m(\bullet\bullet)\ (\circ\circ\bullet)^n(\bullet)\ (\circ\circ\bullet)^p$ m+n+p =i-1

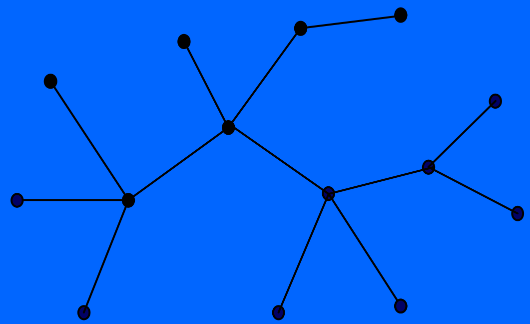
$(\bullet\circ\circ)^a(\bullet\circ)\ (\bullet\circ\circ)^b(\bullet\circ)\ (\bullet\circ\circ)^c(\bullet\bullet)(\circ\circ\bullet)^d$ a+b+c+d =i-2

$(\bullet\circ\circ)^a(\bullet\circ)\ (\bullet\circ\circ)^b(\bullet\bullet)\ (\circ\circ\bullet)^c(\circ\bullet)(\circ\circ\bullet)^d$

$(\bullet\circ\circ)^a(\bullet\bullet)\ (\circ\circ\bullet)^b(\circ\bullet)\ (\bullet\circ\circ)^c(\circ\bullet)(\circ\circ\bullet)^d$

Other Topologies

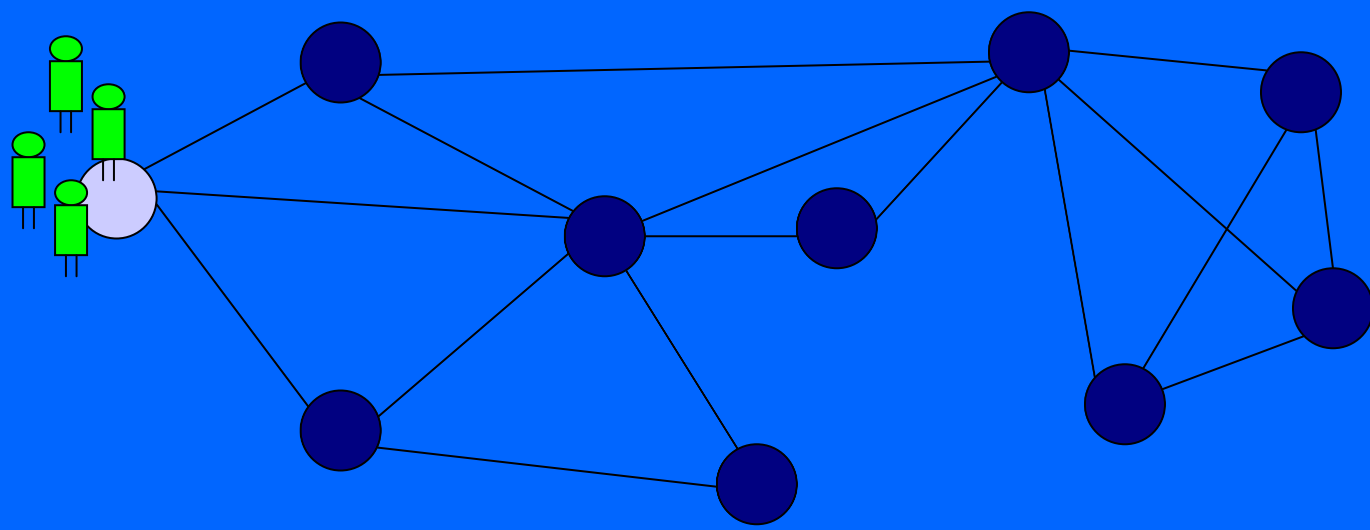




Trees, hypercubes, CCCs, butteflies, DeBrujin …

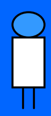
[Journal of Discrete Algorithms: Flocchini et al. 03]

Network Decontamination

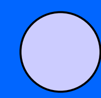


Mobile Agents System

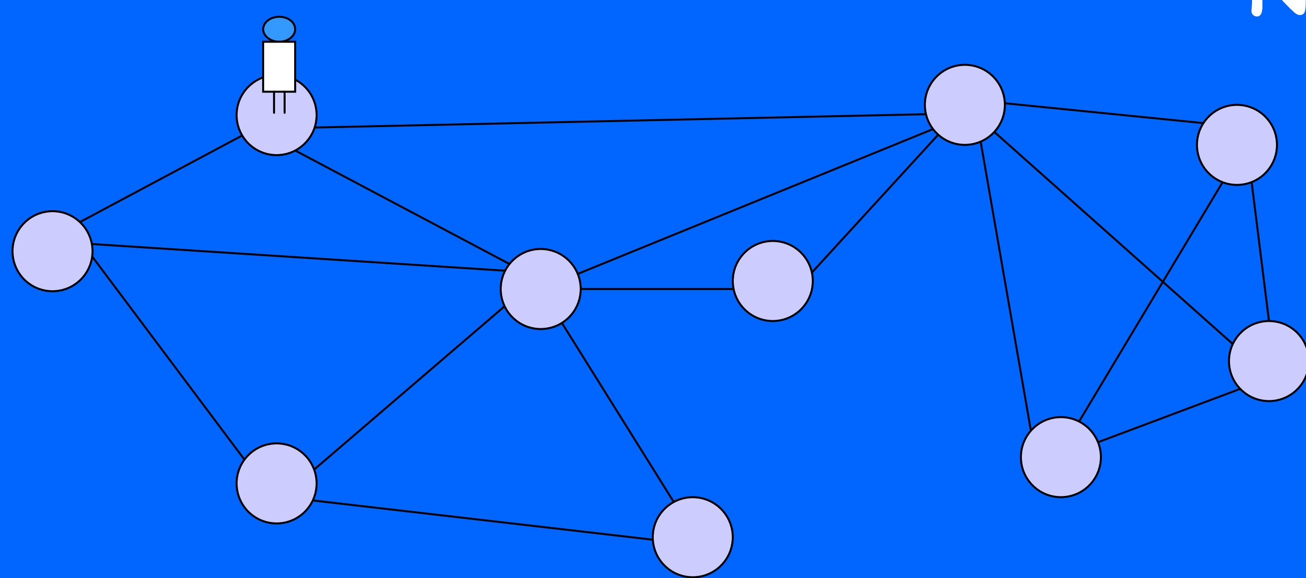
Agents



Sites



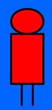
Network

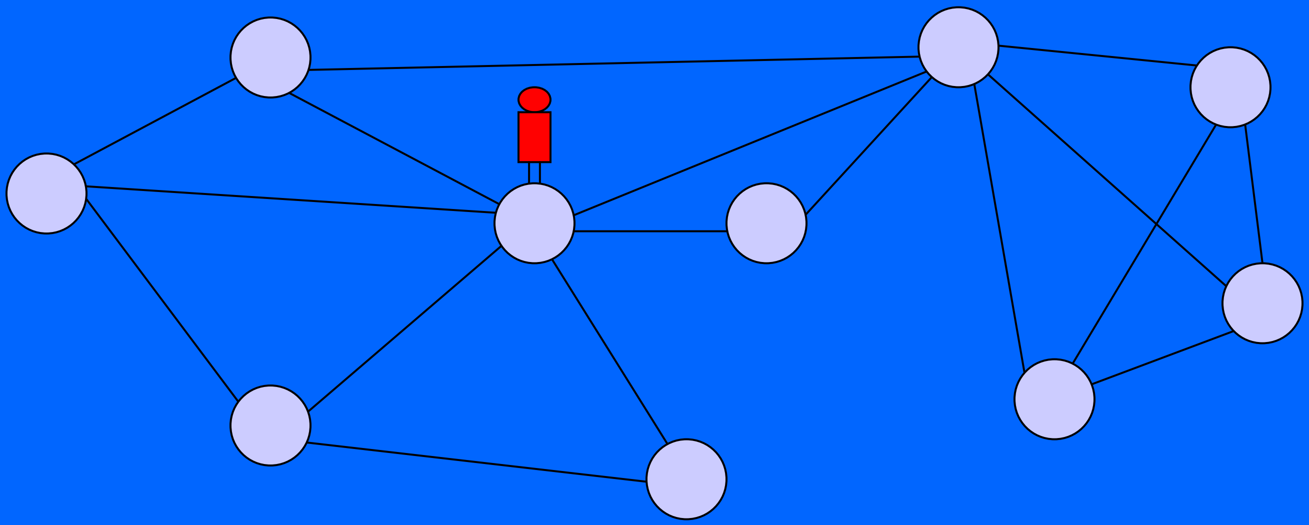


Security Problems

SECURITY PROBLEM

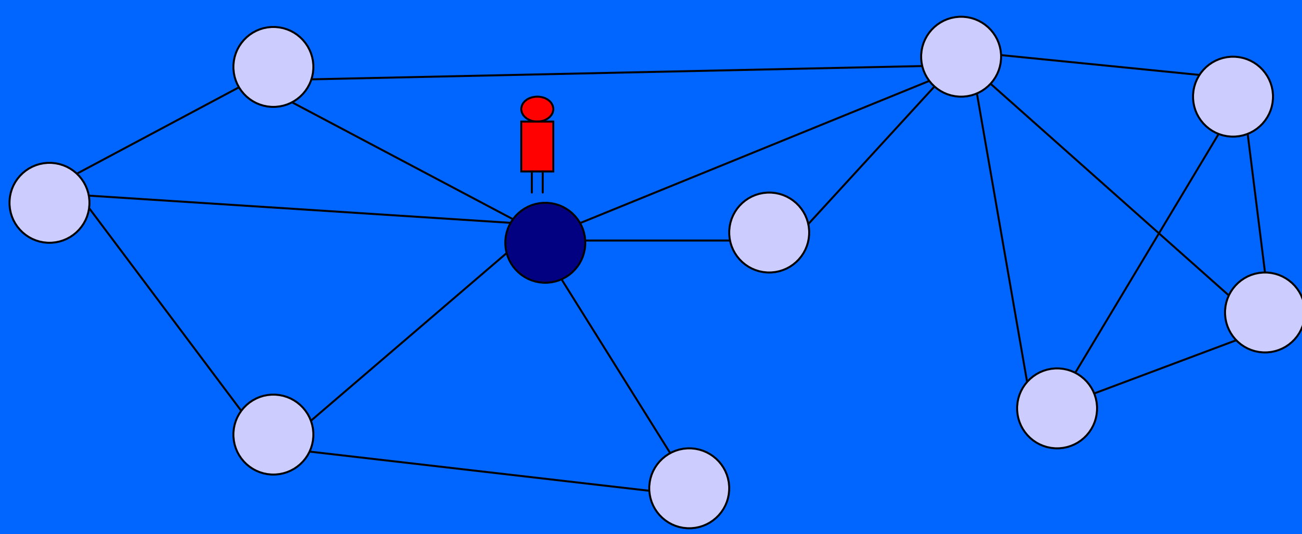
- Harmful Mobile Agent - Virus





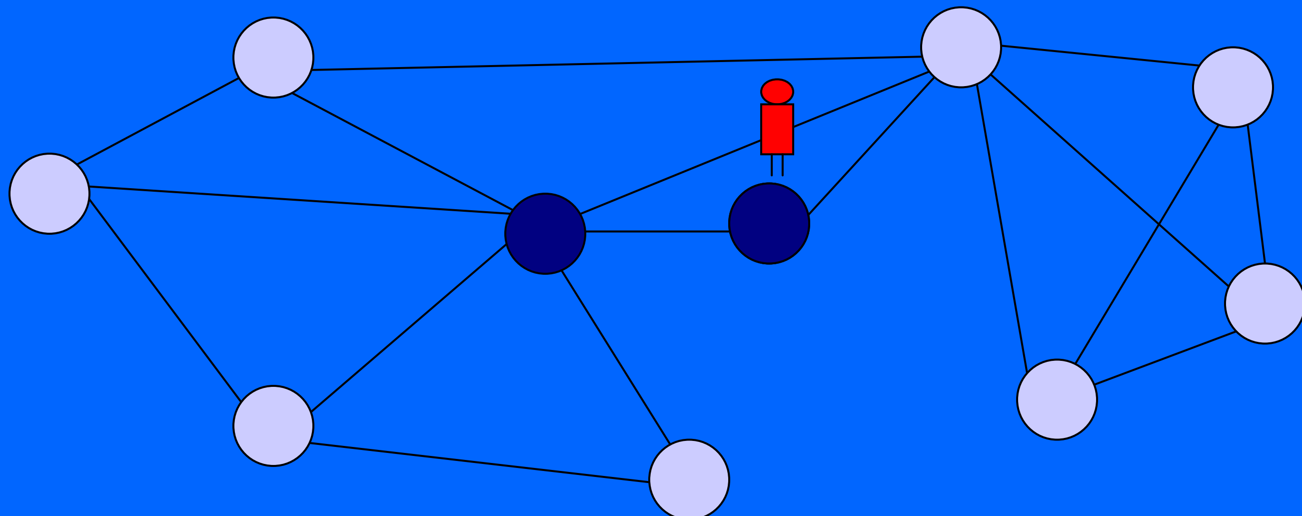
Security Problems

- Harmful Mobile Agent - Virus



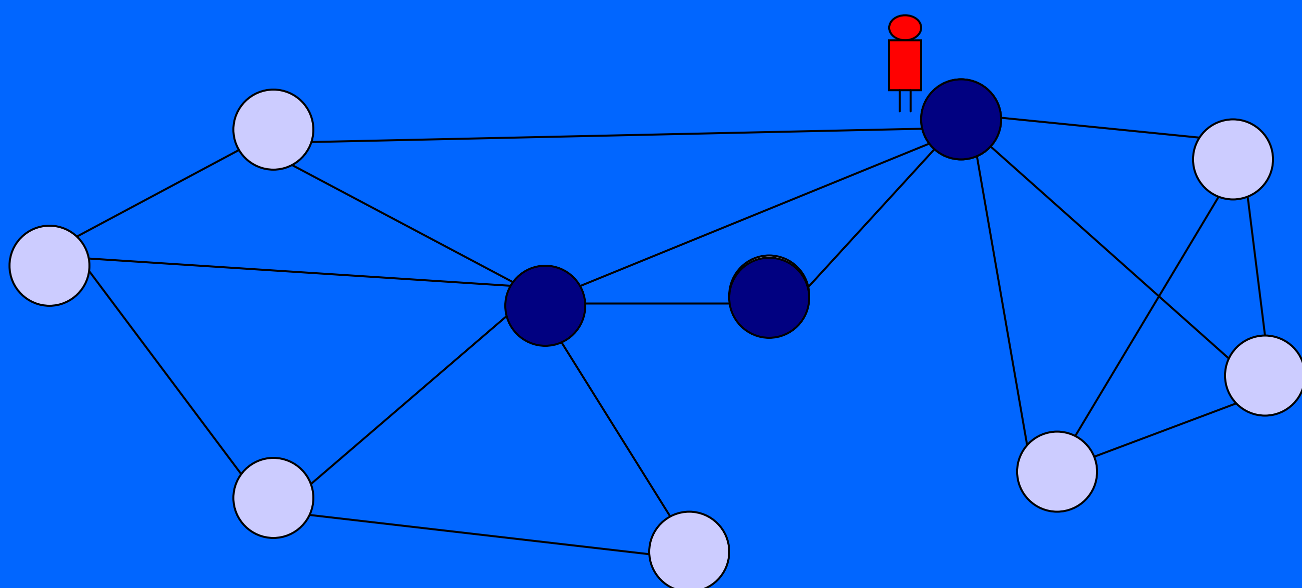
Security Problems

- Harmful Mobile Agent - Virus



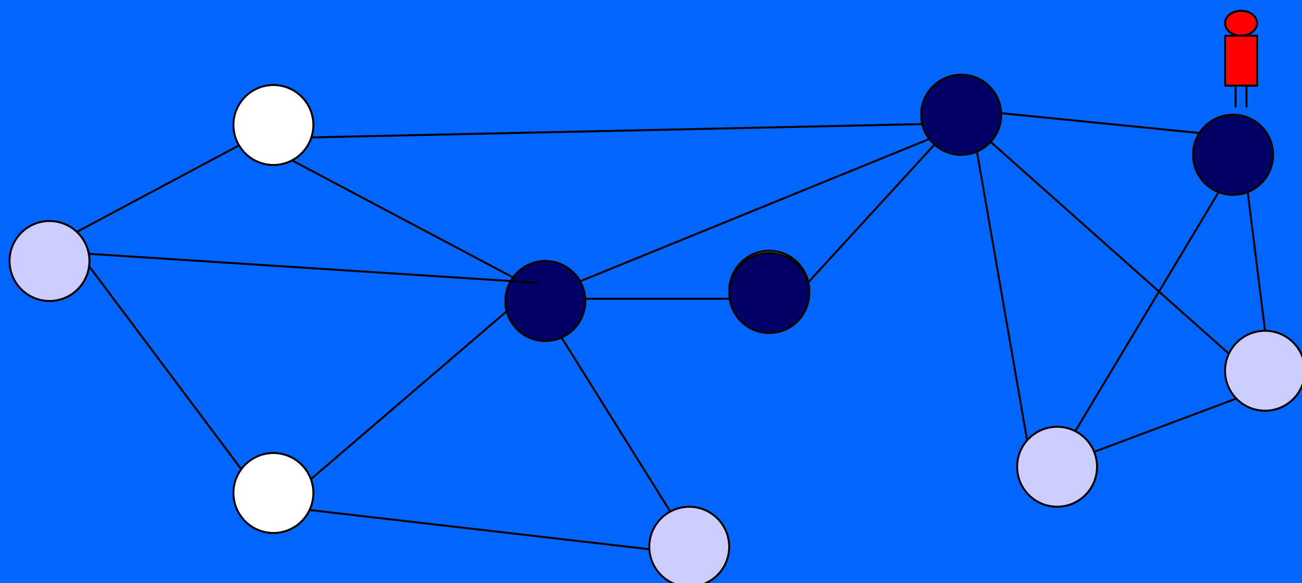
Security Problems

- Harmful Mobile Agent - Virus



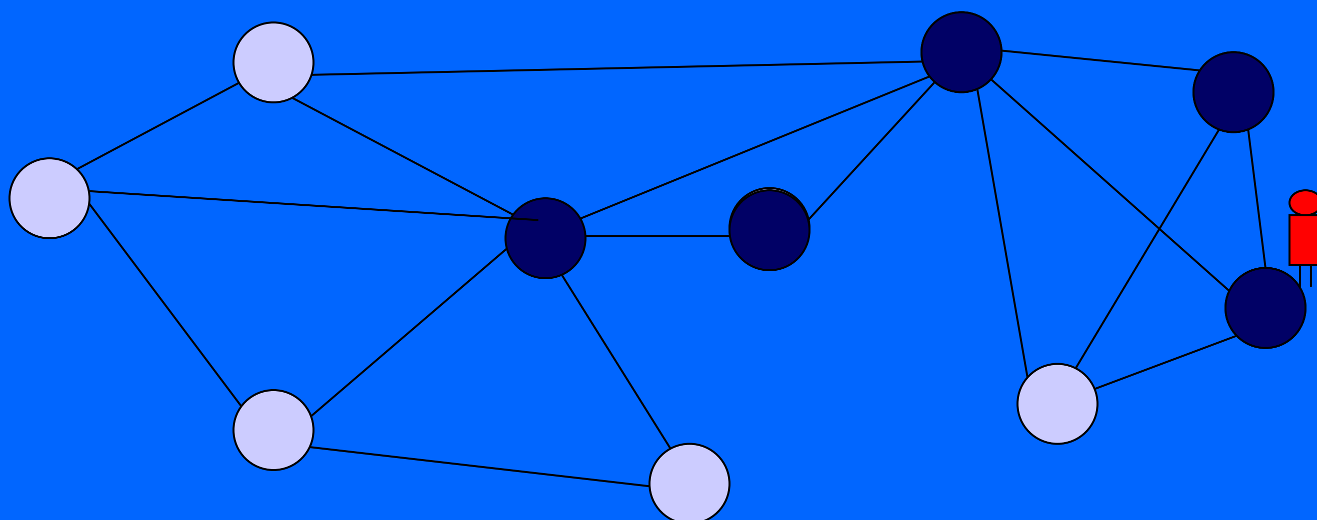
Security Problems

- Harmful Mobile Agent - Virus



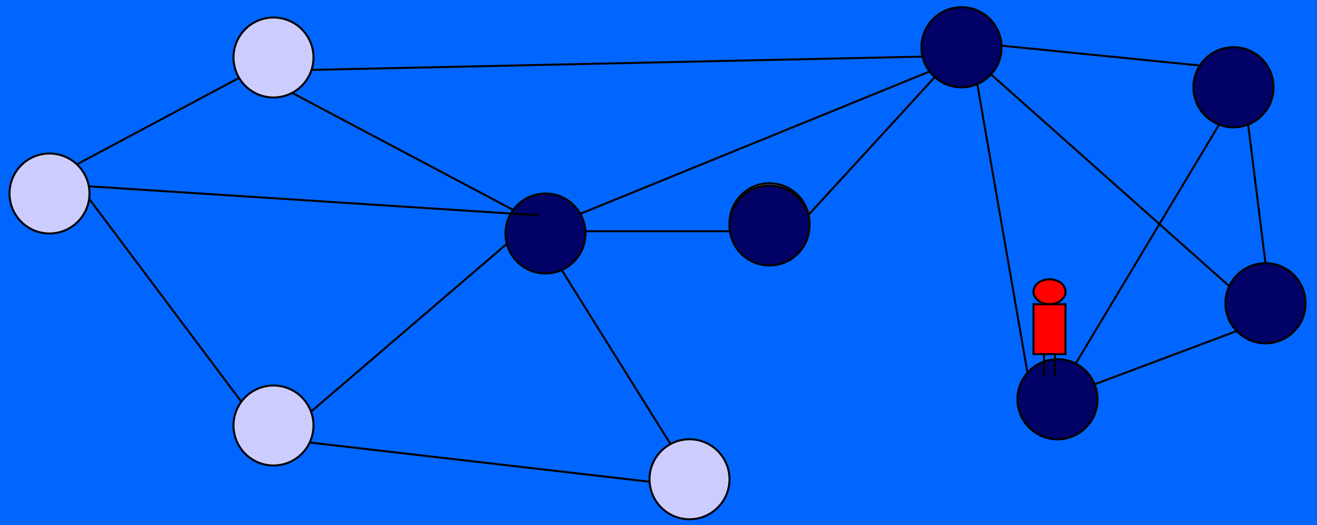
Security Problems

- Harmful Mobile Agent - Virus



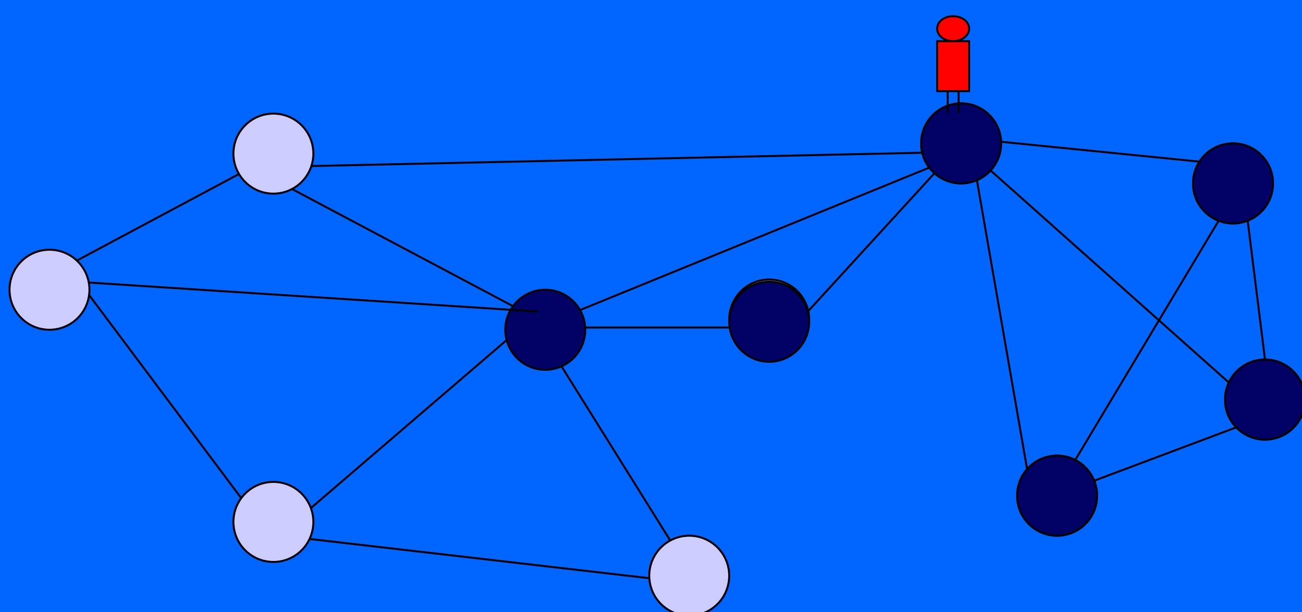
Security Problems

- Harmful Mobile Agent - Virus



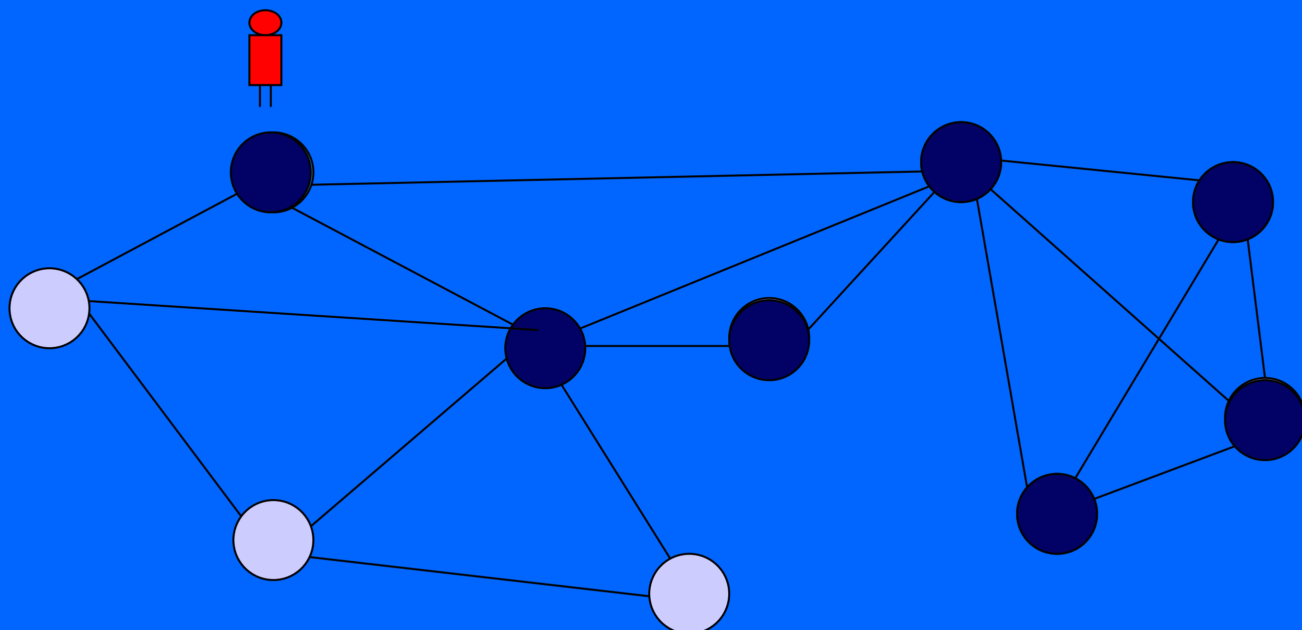
Security Problems

- Harmful Mobile Agent - Virus



Security Problems

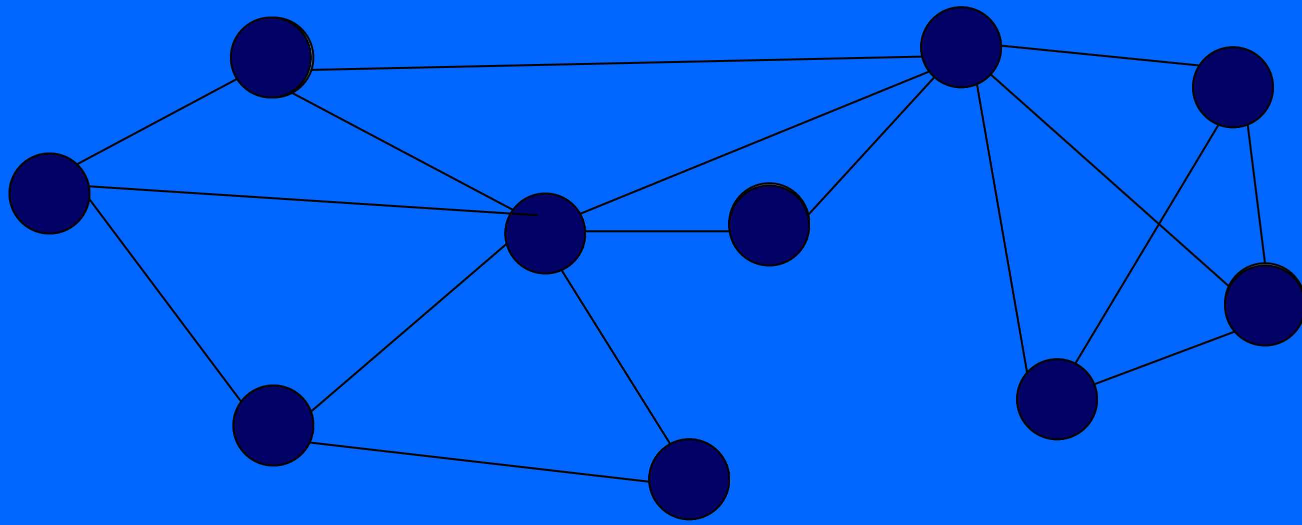
- Harmful Mobile Agent - Virus



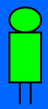
Security Problems

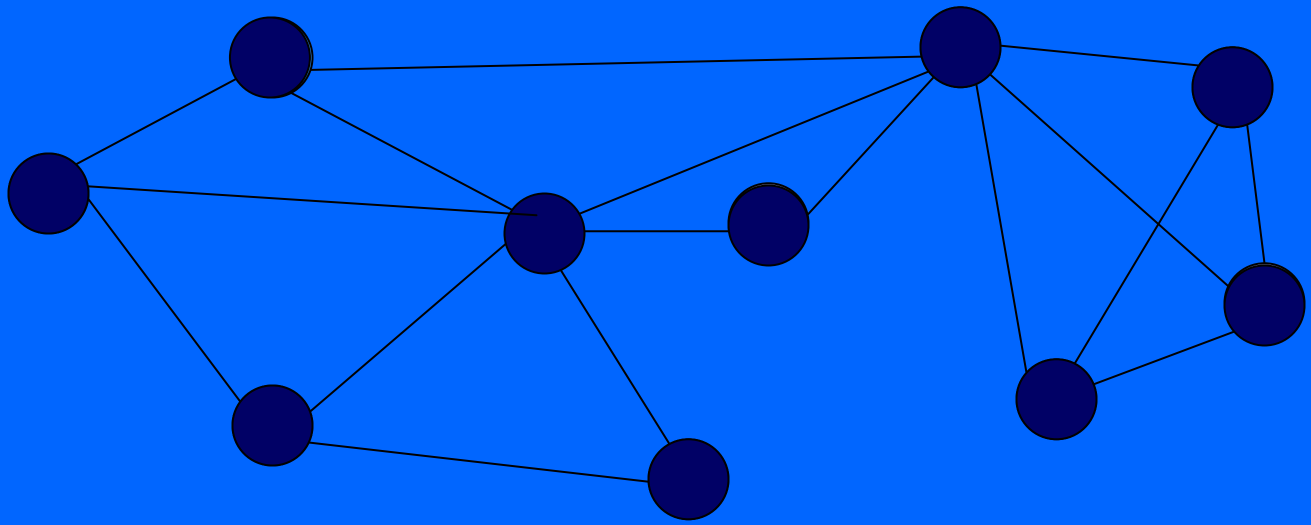
- Harmful Mobile Agent - Virus

  Network is *CONTAMINATED*



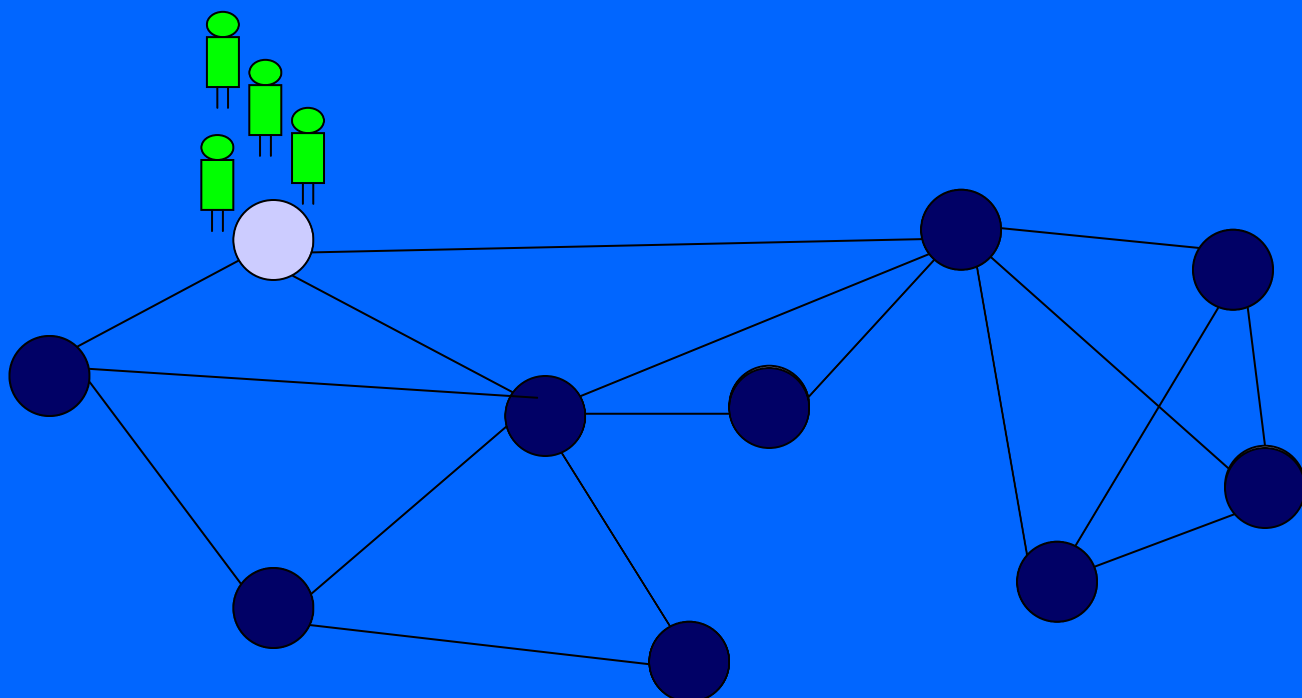
DECONTAMINATION

- Team of System Agents - *Cleaners* 



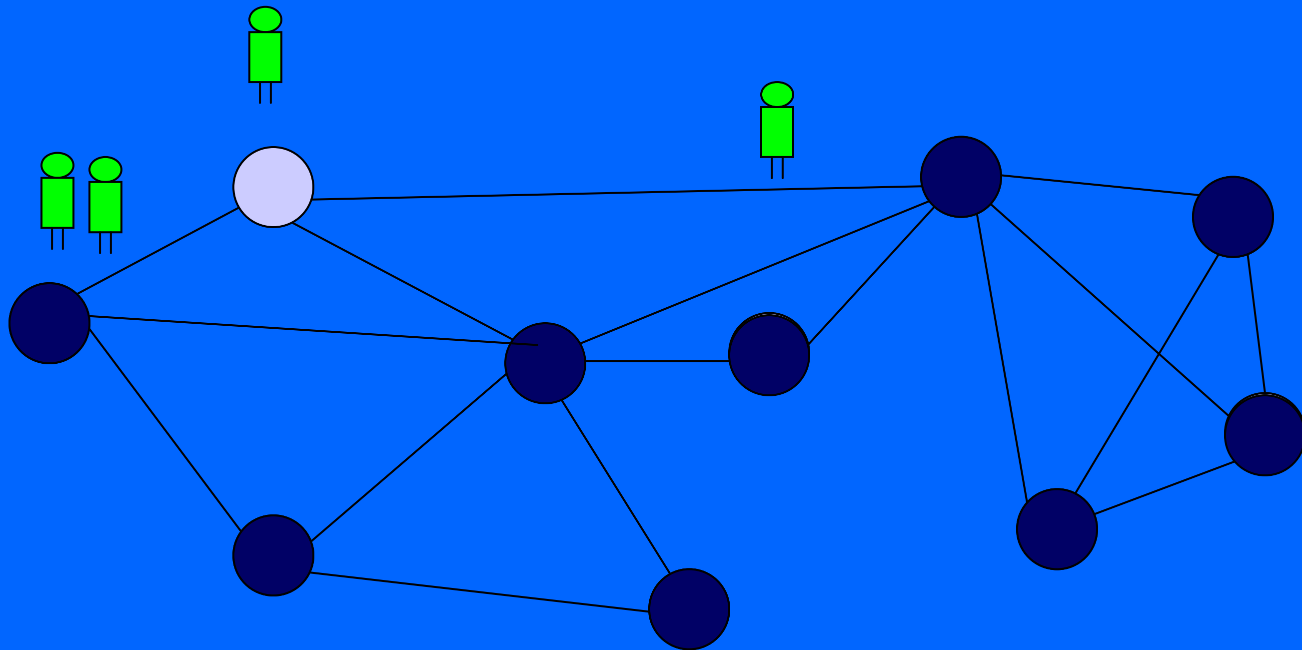
DECONTAMINATION

- Team of System Agents - *Cleaners*



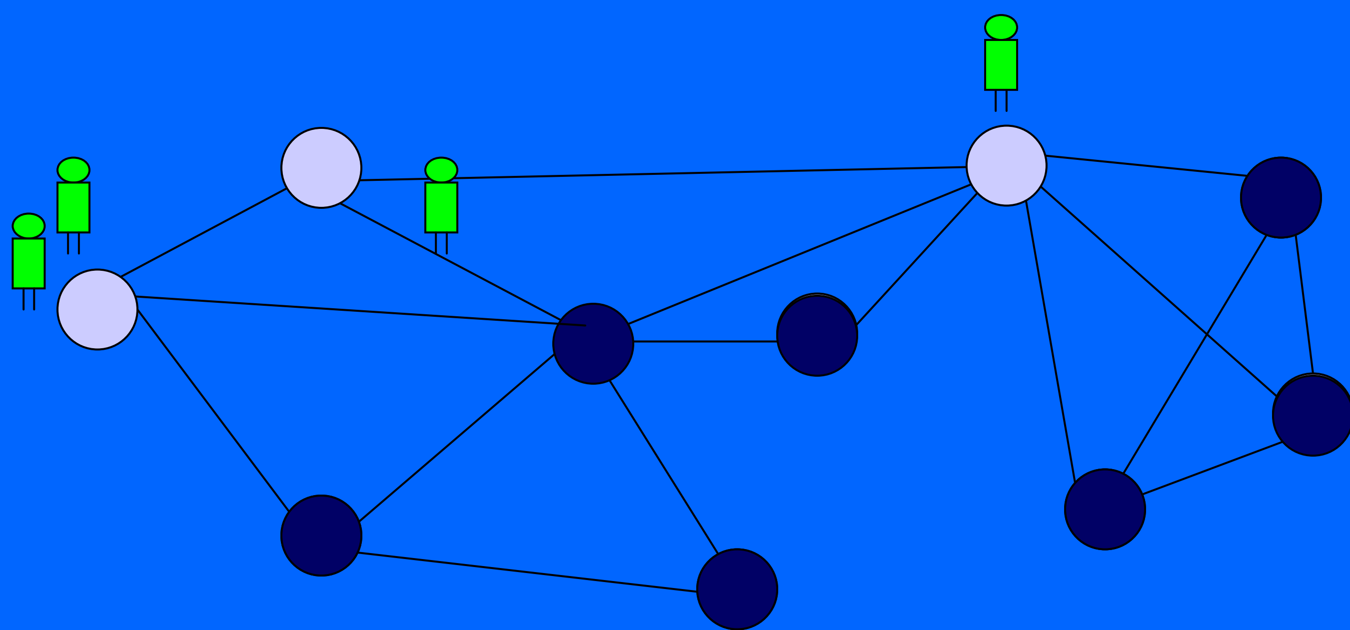
DECONTAMINATION

- Team of System Agents - *Cleaners*



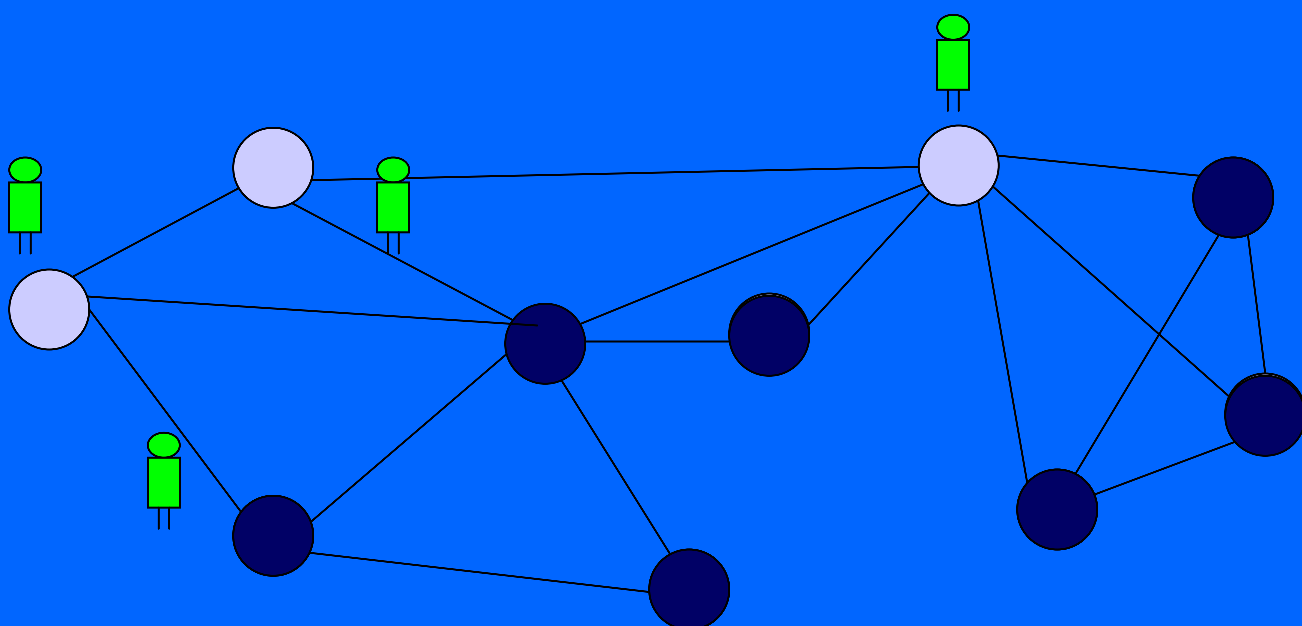
DECONTAMINATION

- Team of System Agents - *Cleaners*



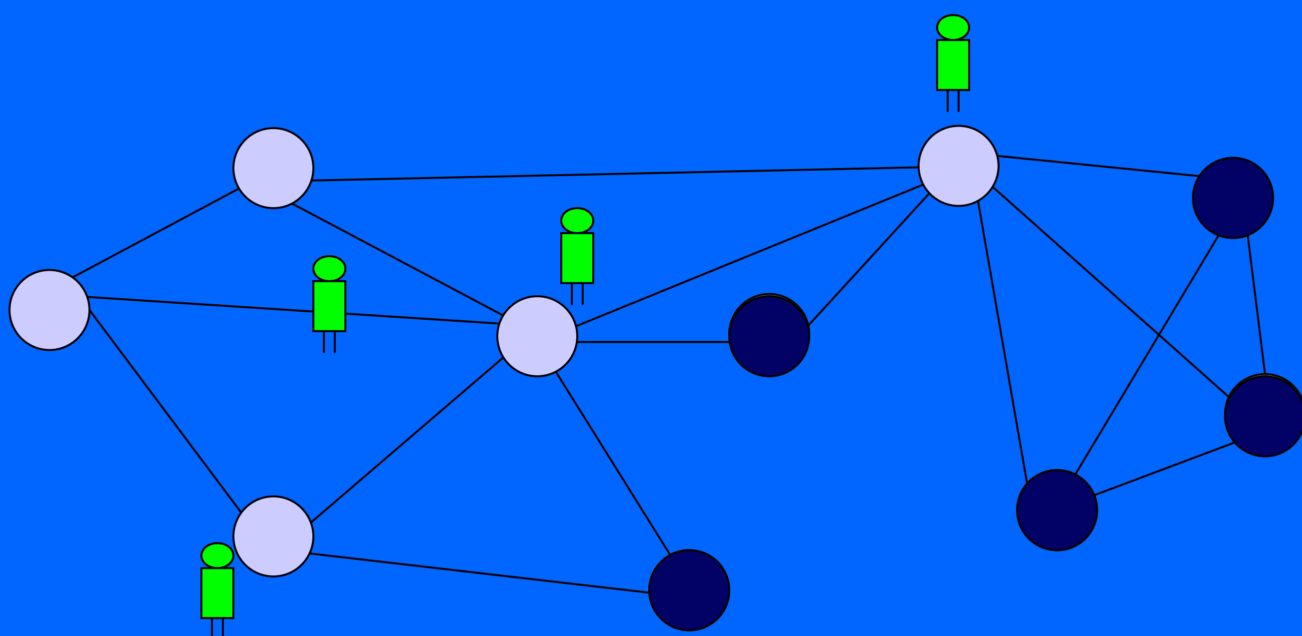
DECONTAMINATION

- Team of System Agents - *Cleaners*



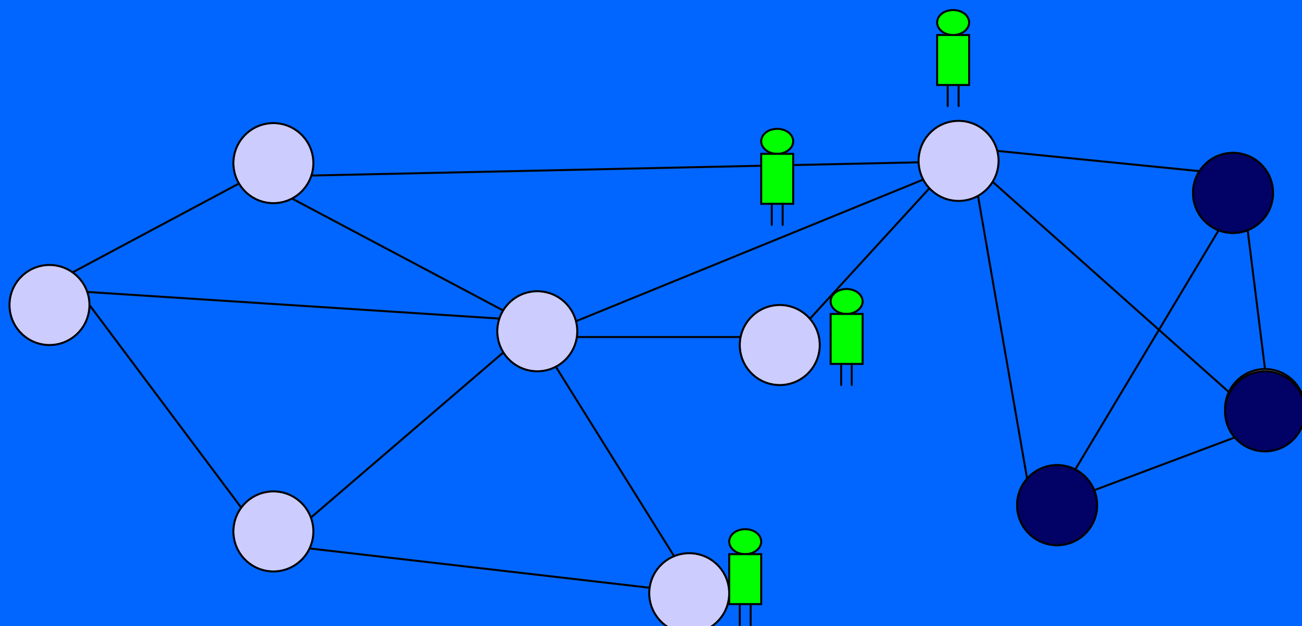
DECONTAMINATION

- Team of System Agents - *Cleaners*



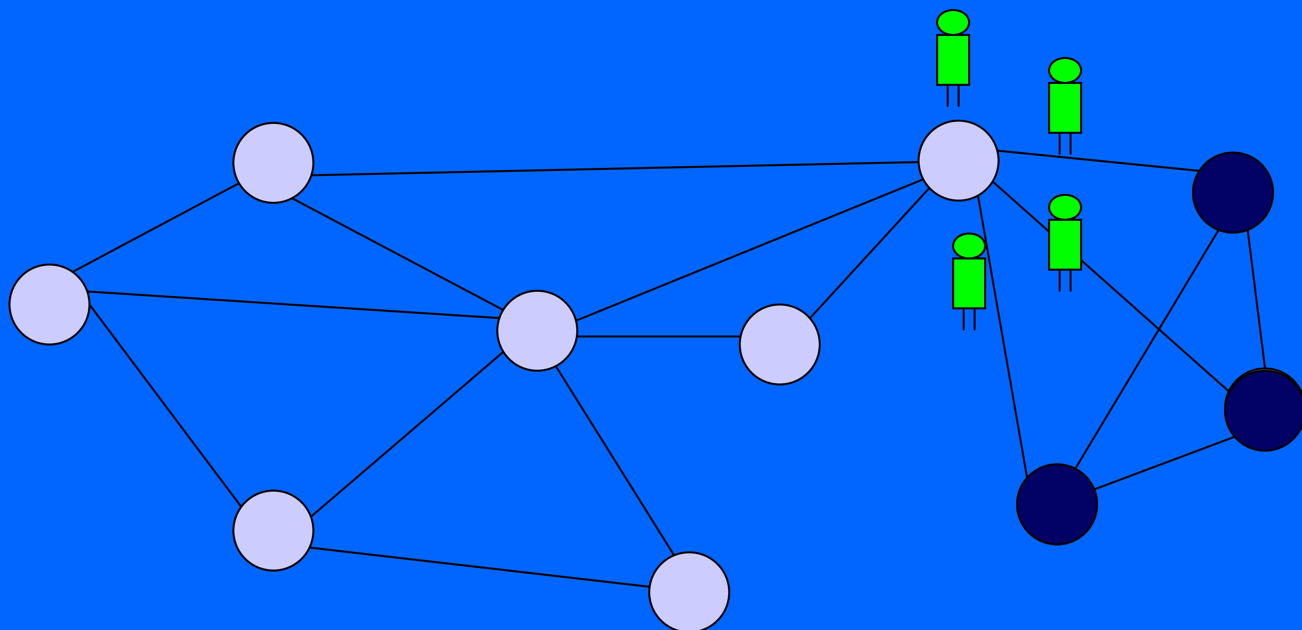
DECONTAMINATION

- Team of System Agents - *Cleaners*



DECONTAMINATION

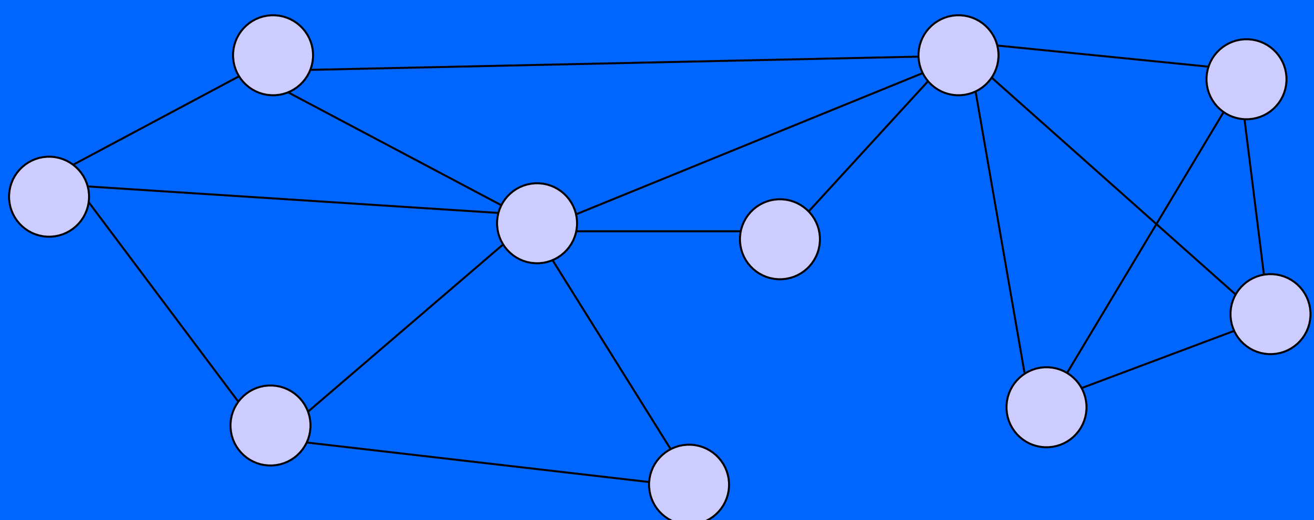
- Team of System Agents - *Cleaners*



DECONTAMINATION

- Team of System Agents - *Cleaners*

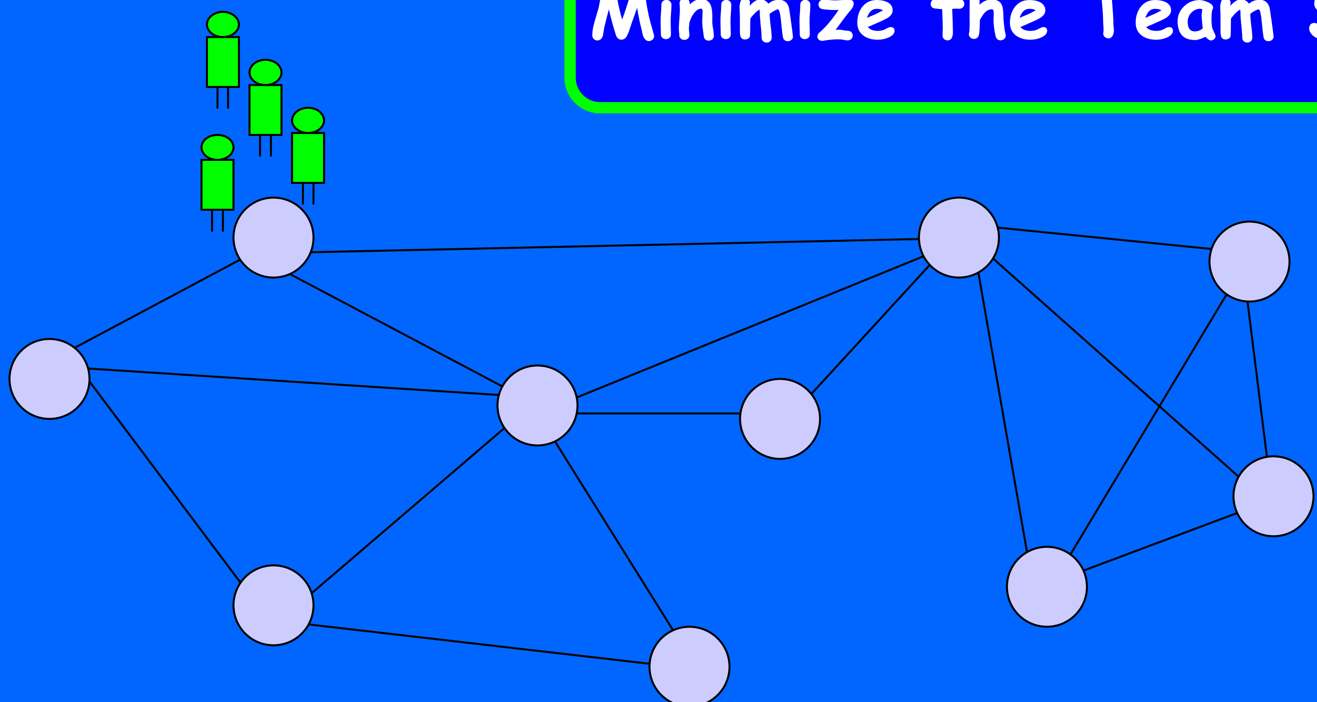
  Network is *DECONTAMINATED*



DECONTAMINATION

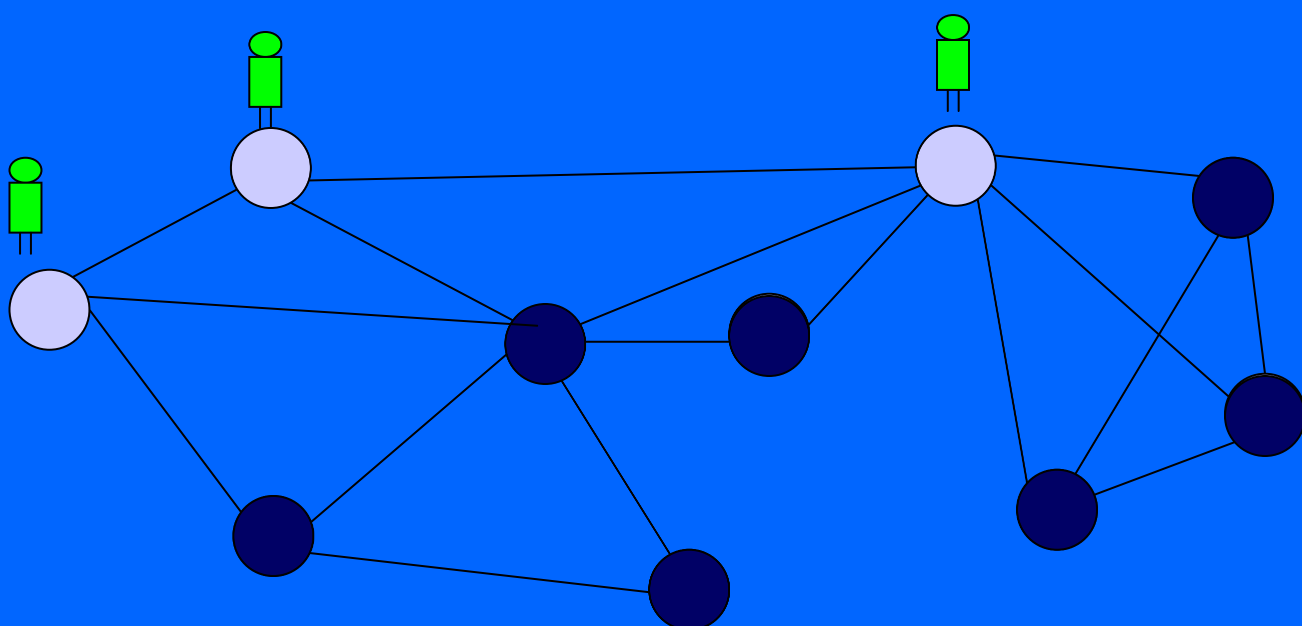
- Team Size *(how many agents?)*
- Cleaning Strategy *(the algorithm)*
- Number of Moves

Minimize the Team Size



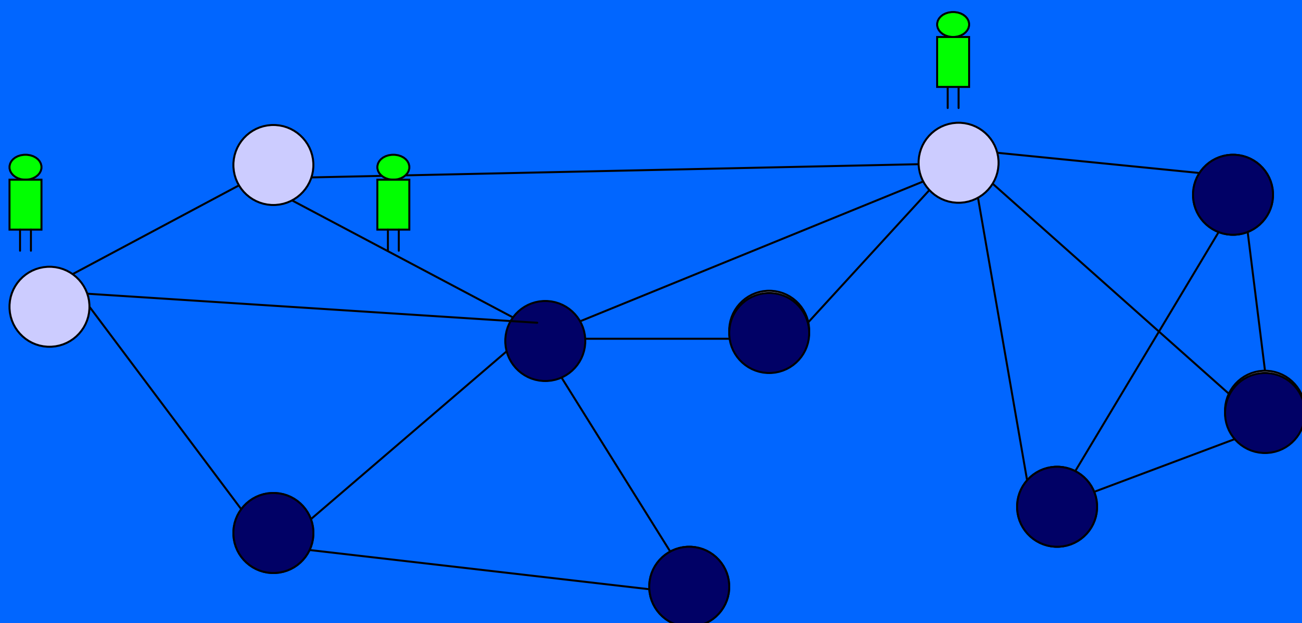
DECONTAMINATION

- PROBLEM:



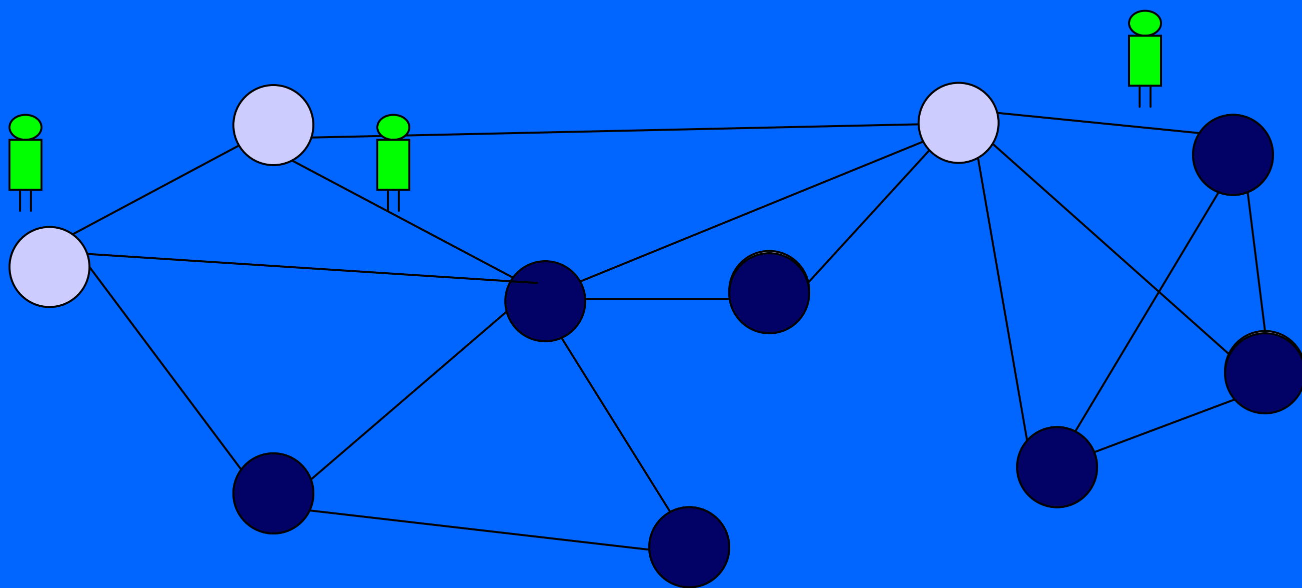
DECONTAMINATION

- PROBLEM:



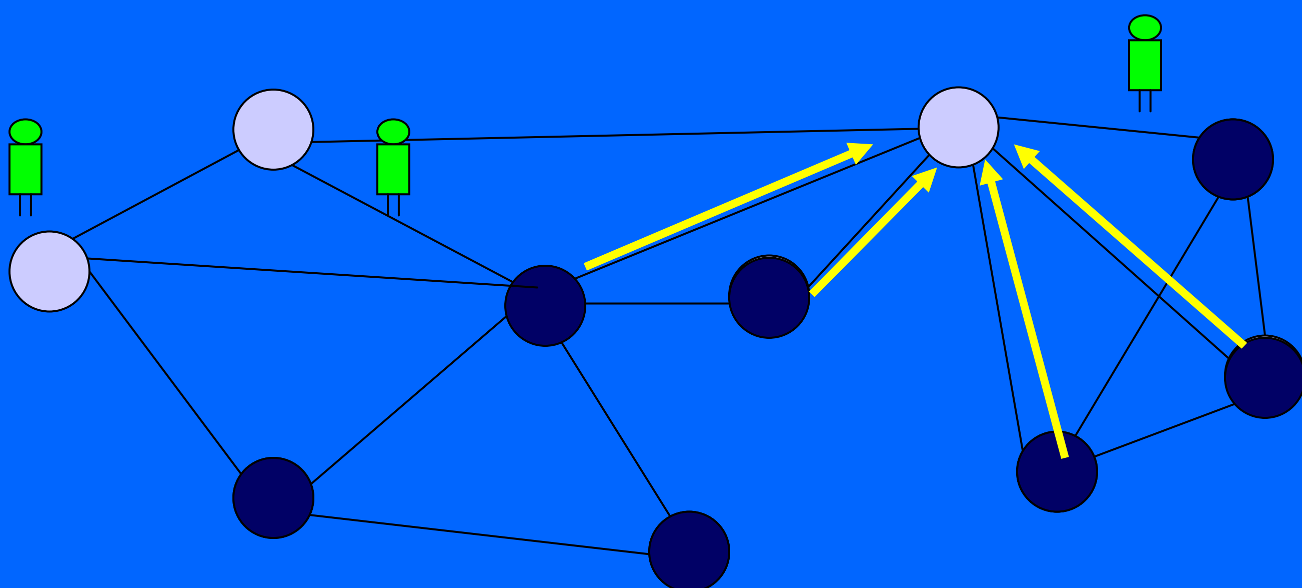
DECONTAMINATION

- PROBLEM:



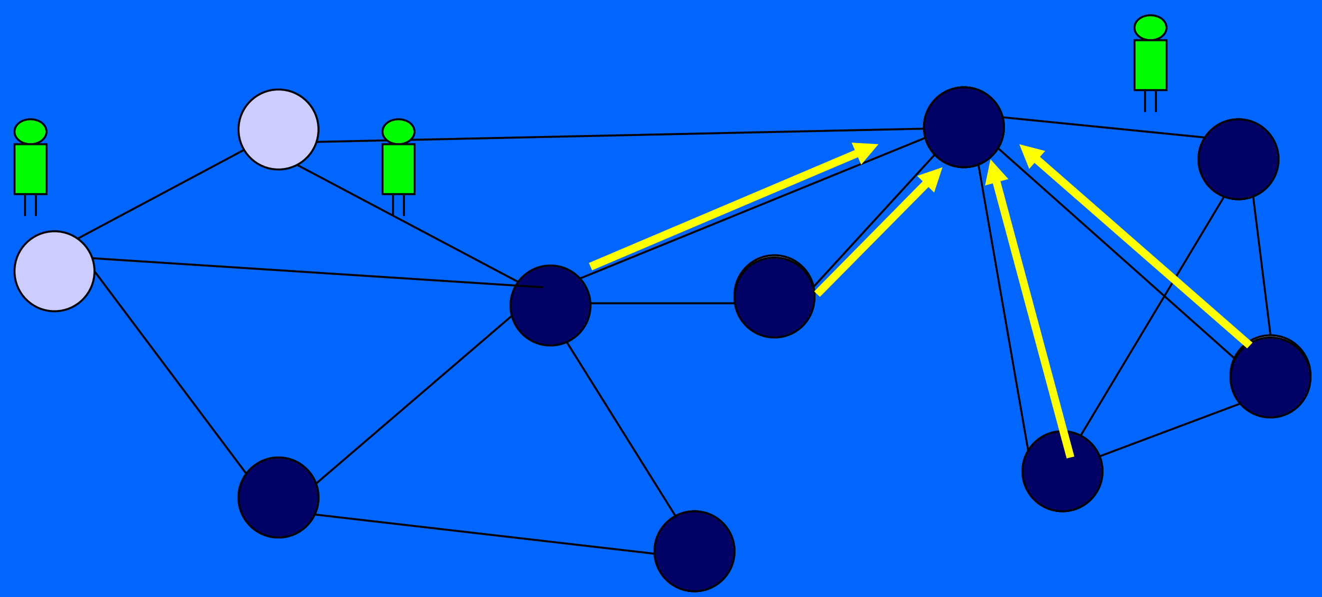
DECONTAMINATION

- PROBLEM: ***RECONTAMINATION***



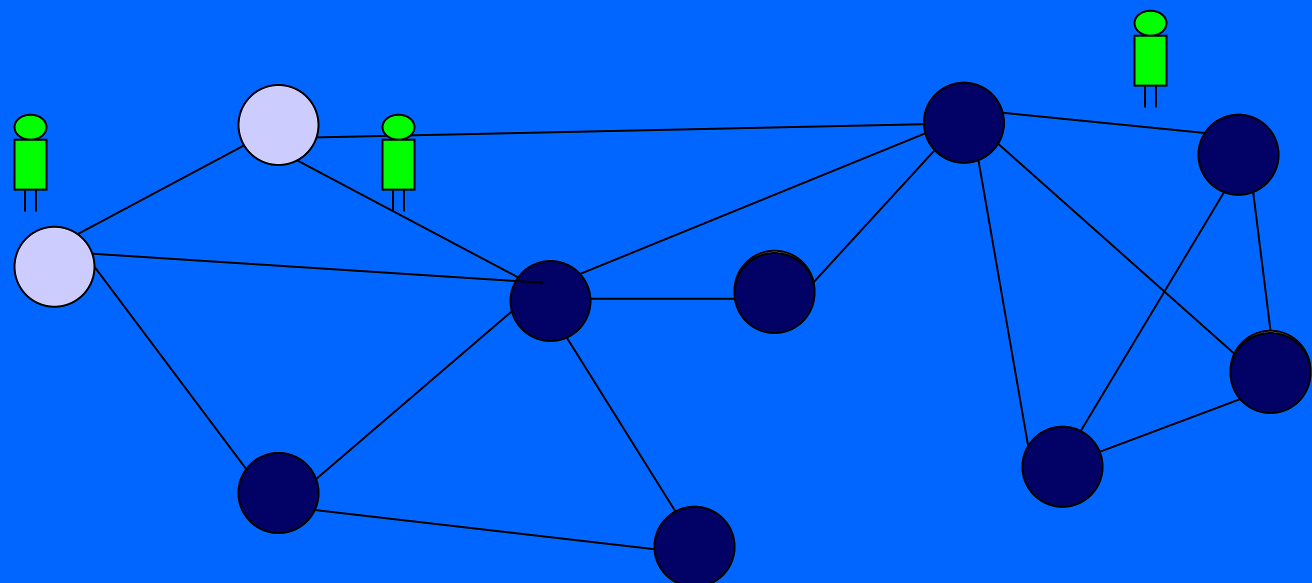
DECONTAMINATION

- PROBLEM: *RECONTAMINATION*



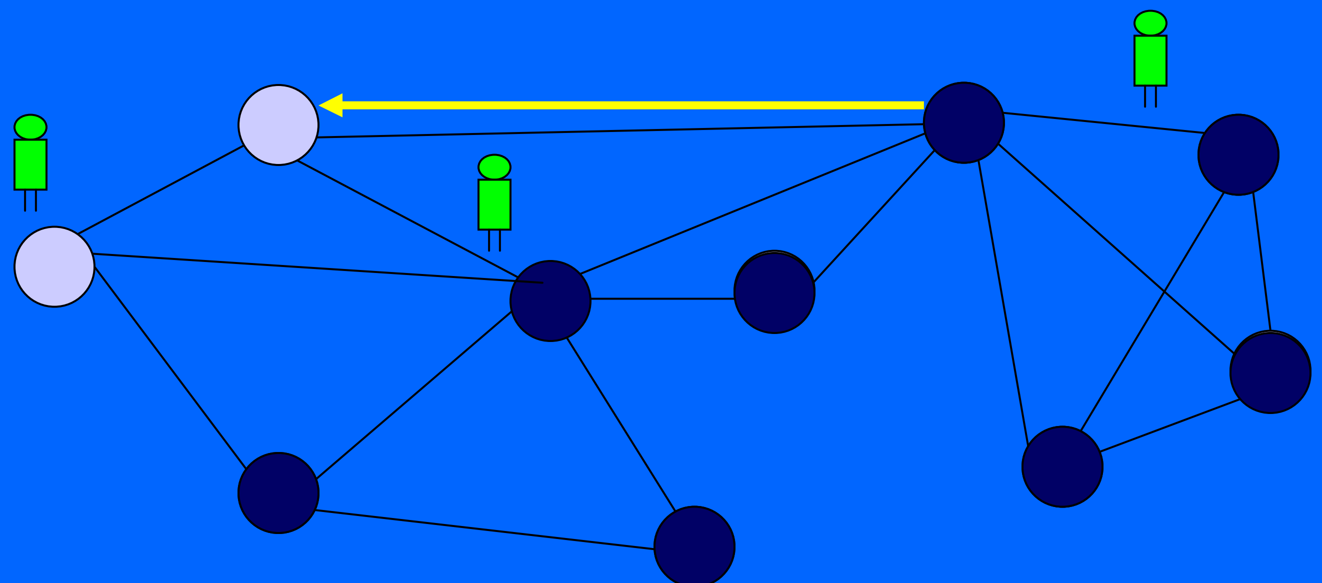
DECONTAMINATION

- PROBLEM: *RECONTAMINATION*



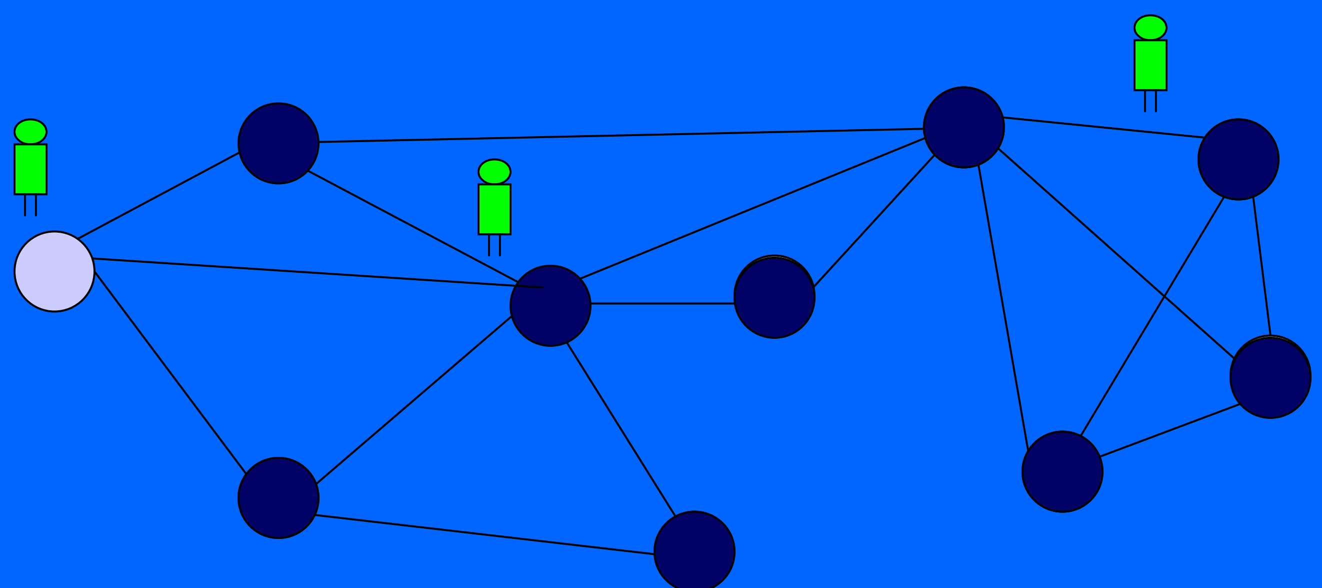
DECONTAMINATION

- PROBLEM: *RECONTAMINATION*



DECONTAMINATION

- PROBLEM: *RECONTAMINATION*



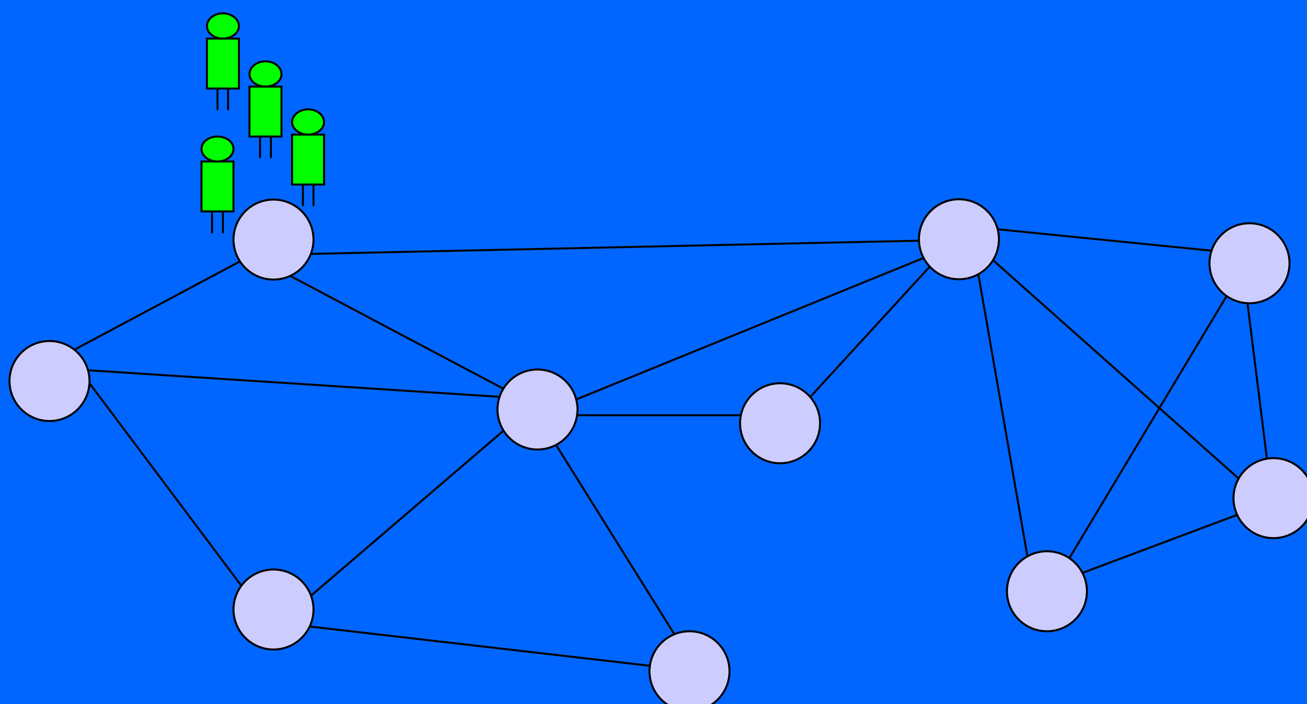
DECONTAMINATION

STRATEGY

Must terminate after a finite number of moves

Must avoid RECONTAMINATION

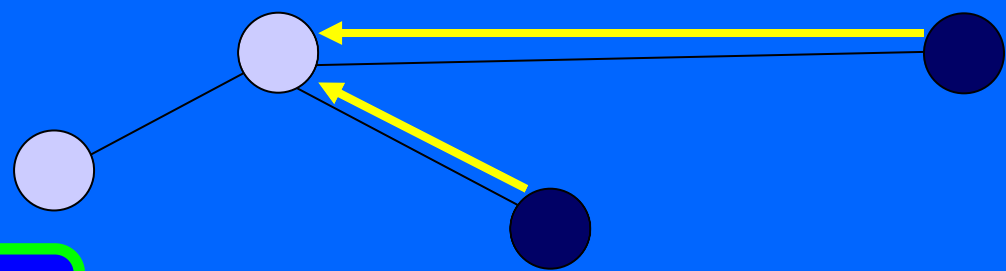
(monotone)



Dynamics described by two concurrent processes:

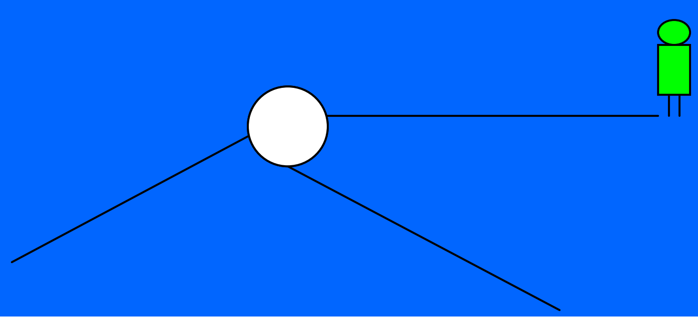
Contamination local, static (like for DYNAMOS)

A clean but unguarded site is contaminated if a certain number of its neighbours (threshold) are contaminated



Decontamination movement

A contaminated node becomes clean if an agent moves on it



Goal

Contamination dynamics

and

Decontamination dynamics

Monocromatic fixed point (like for dynamos): all nodes must be clean

The minimum size of the team
Depends on the topology and on the

RECONTAMINATION RULE

CONTIGUOUS SEARCH problem

-Trees

Barriere,Flocchini,Fraignaud,Santoro (SPAA'02)

- Hypercubes

Flocchini,Huang,F.L.Luccio (Networks 07)

- Meshes

Flocchini,F.L.Luccio,Song (CIC'05)

- Chordal Rings and Tori

Flocchini,Huang,F.L.Luccio (Int. J. of Foundation of Computer Science, 07)

- Outerplanar Graphs

Fomin,Thilikos,Todineau (ICGT'05)

-Other

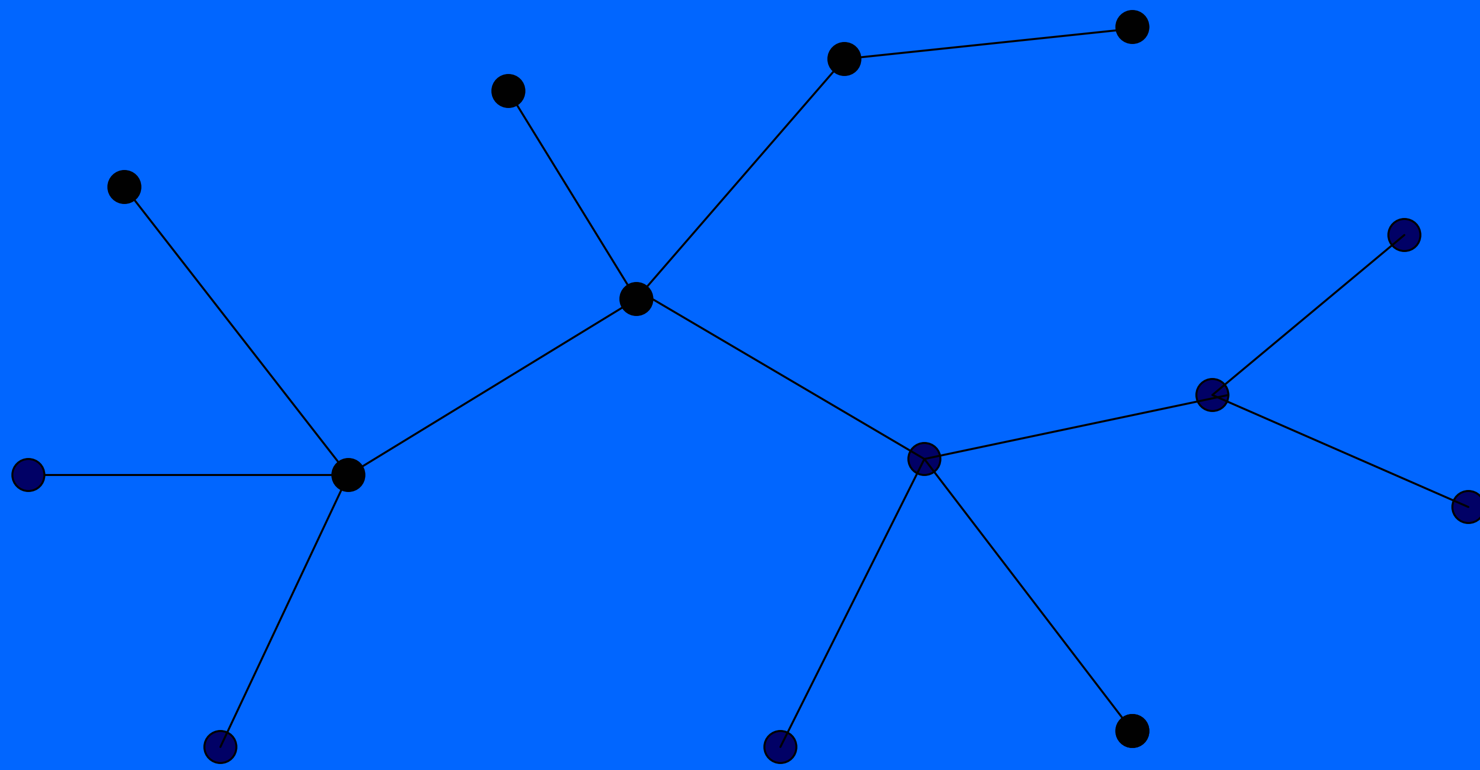
Barriere,Fraignaud,Santoro,Thilikos (WG'03)

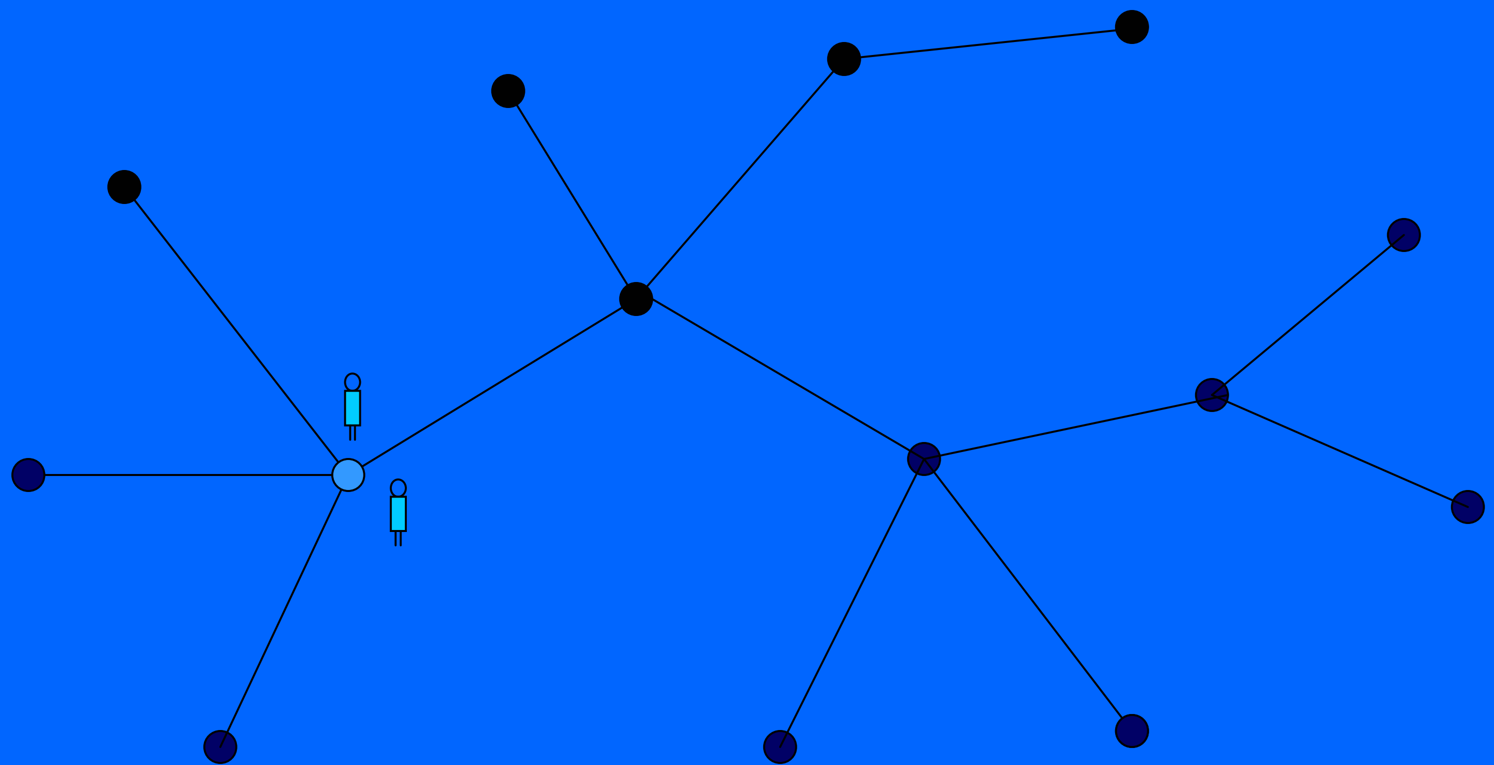
Nisse (SIROCCO 07)

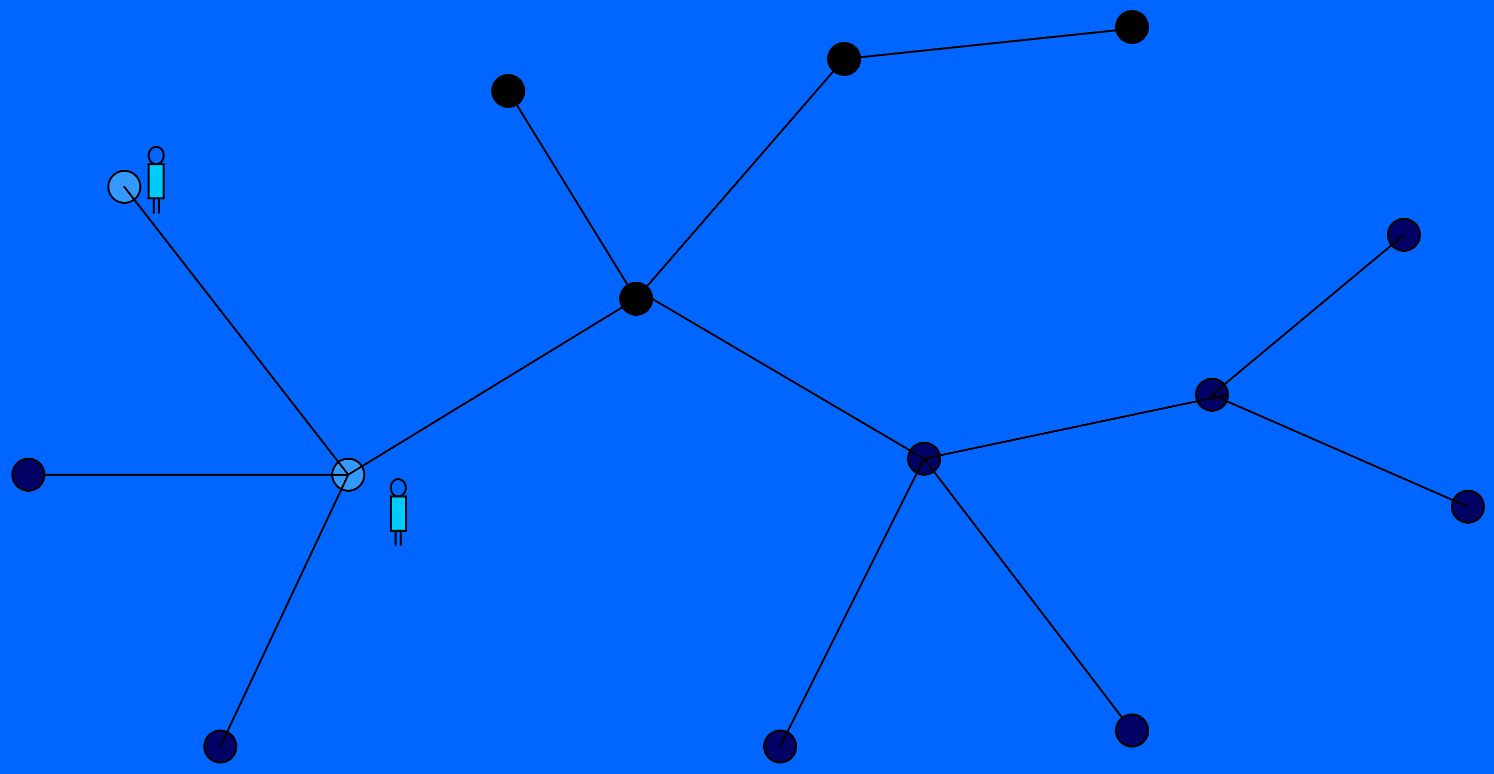
A single contaminated neighbour suffices for recontamination

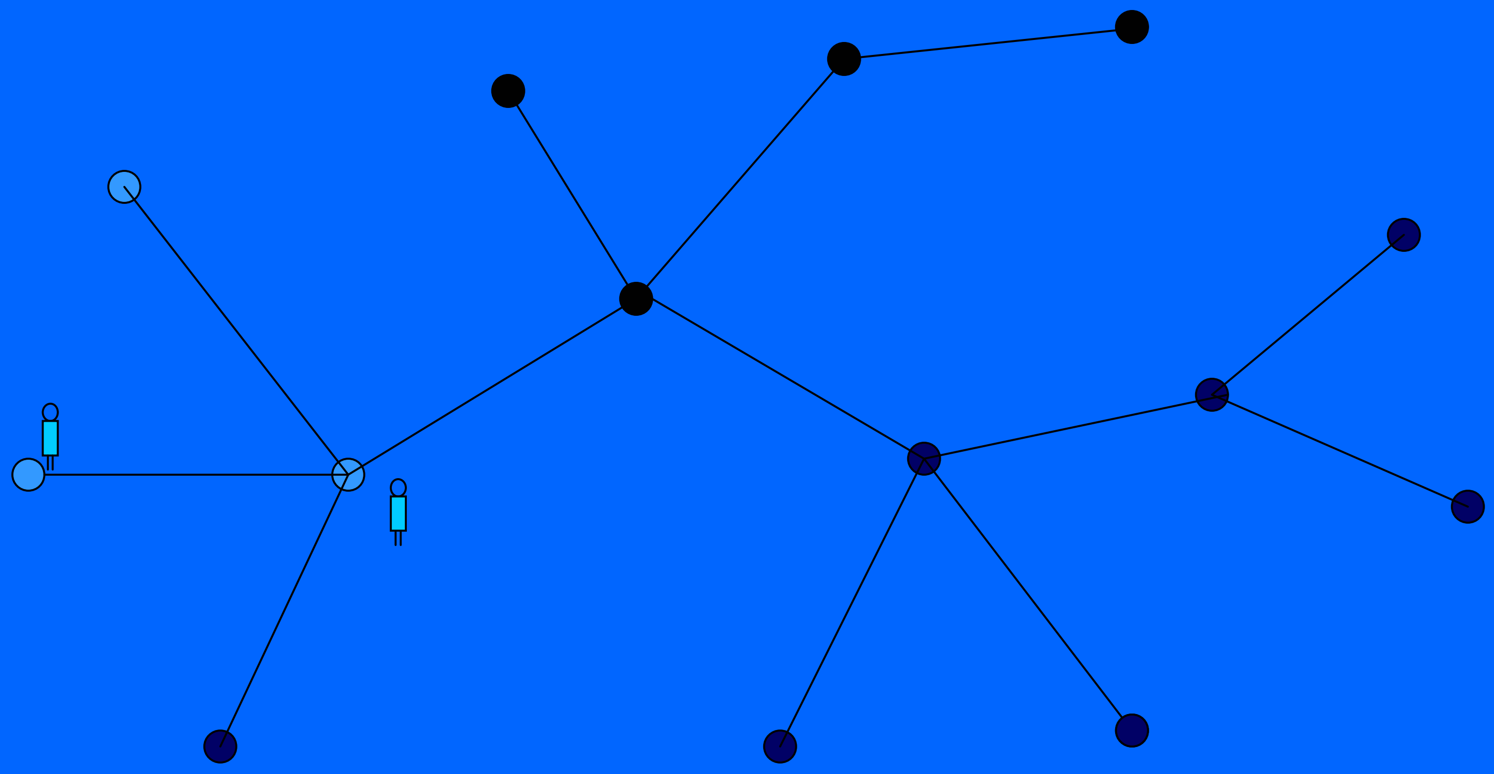
Example:

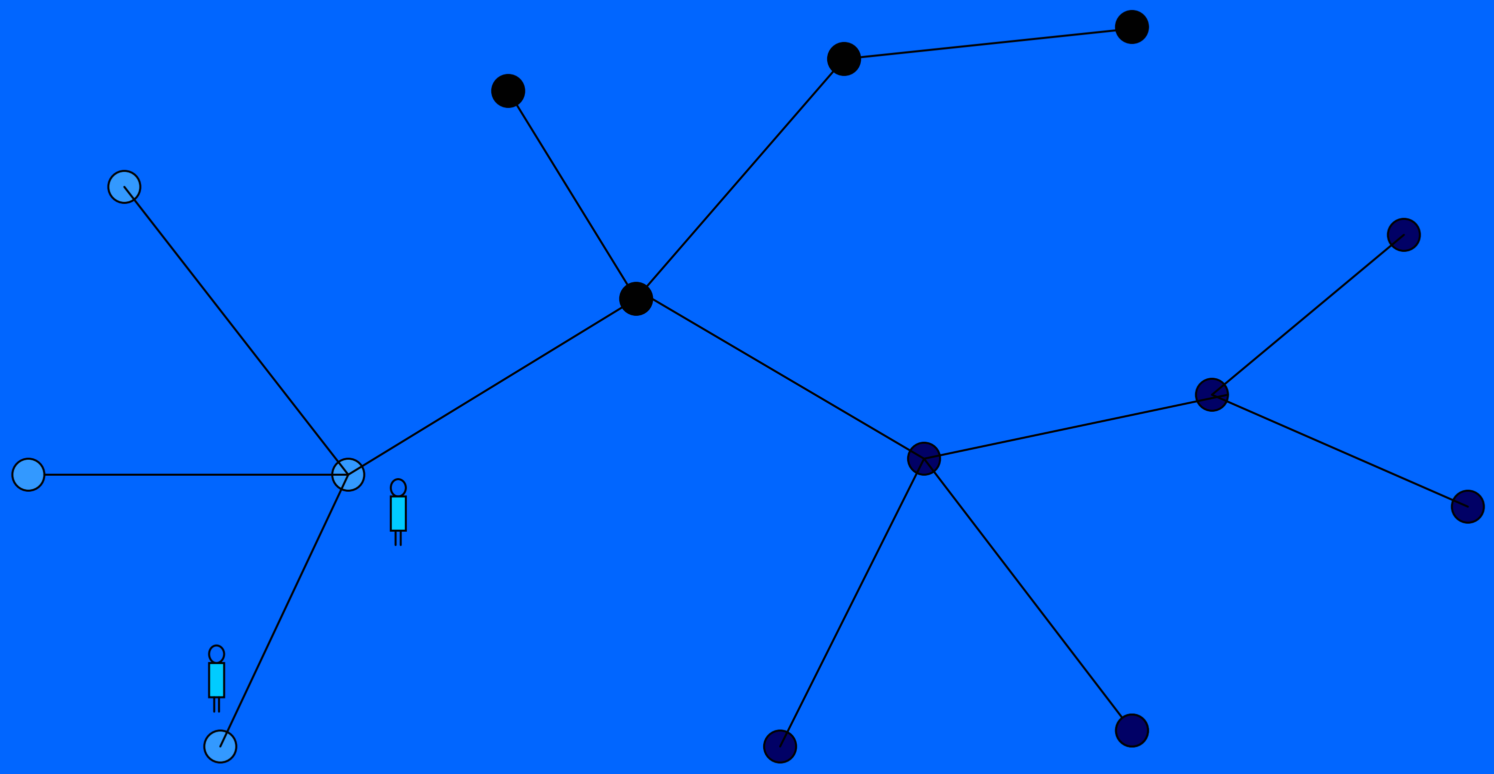
How many agents can clean it ?

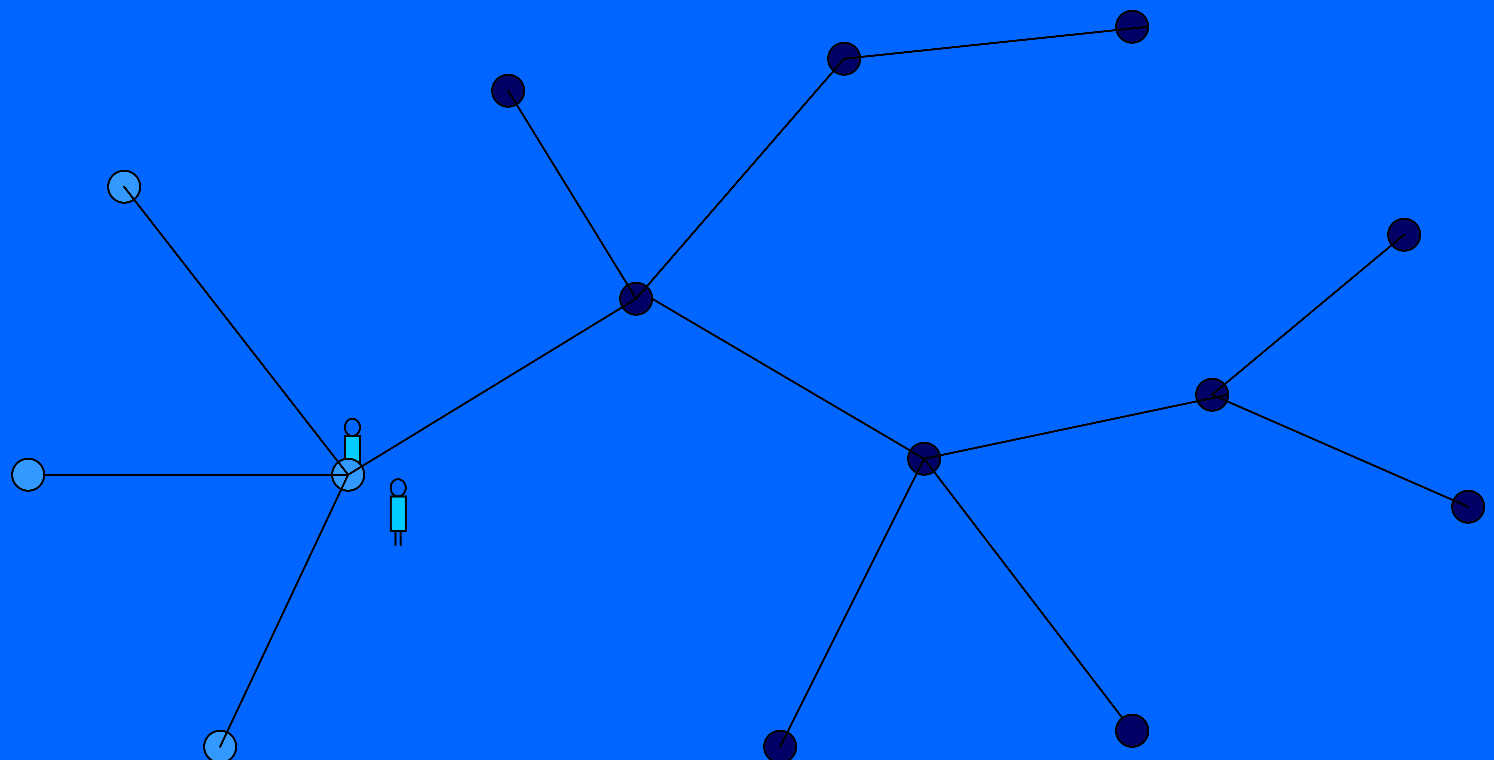


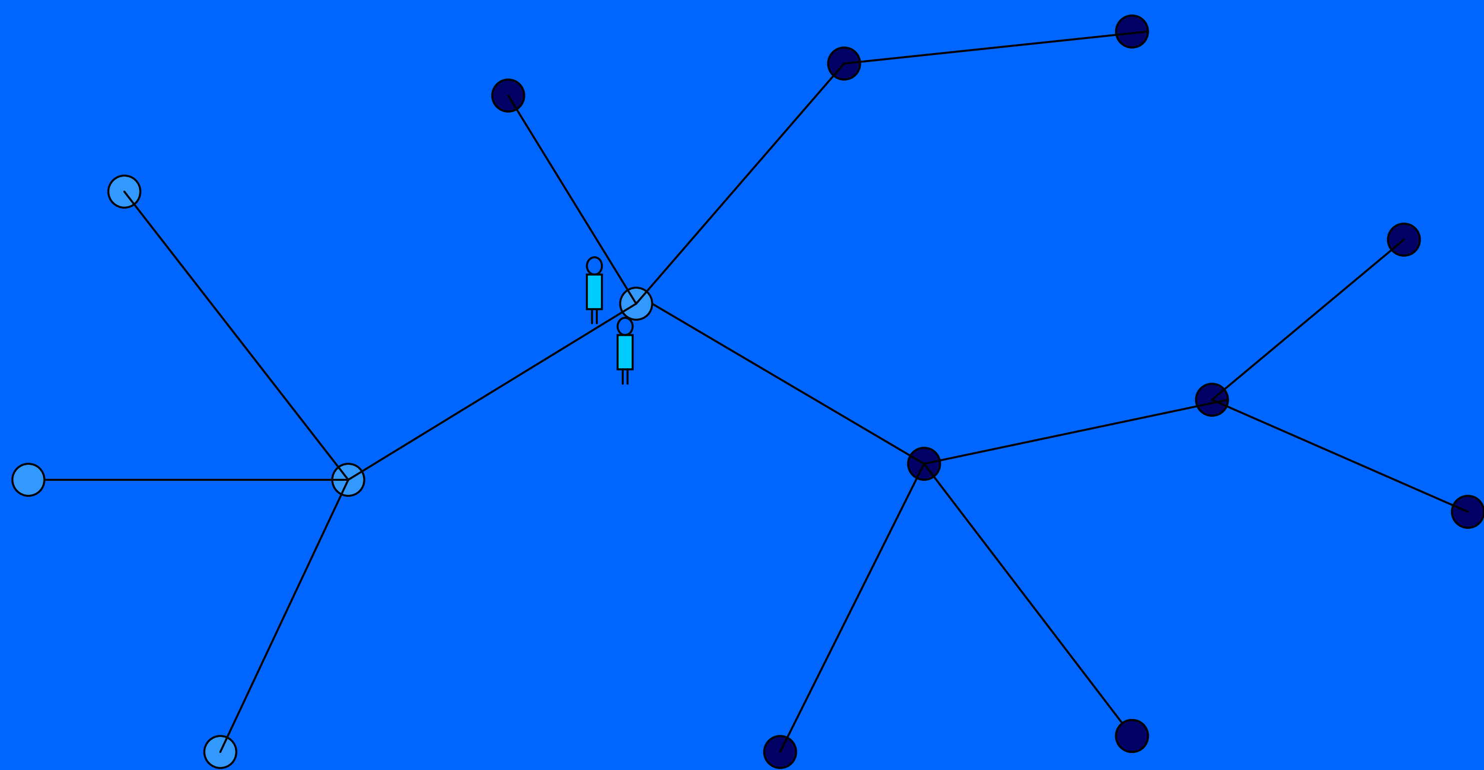


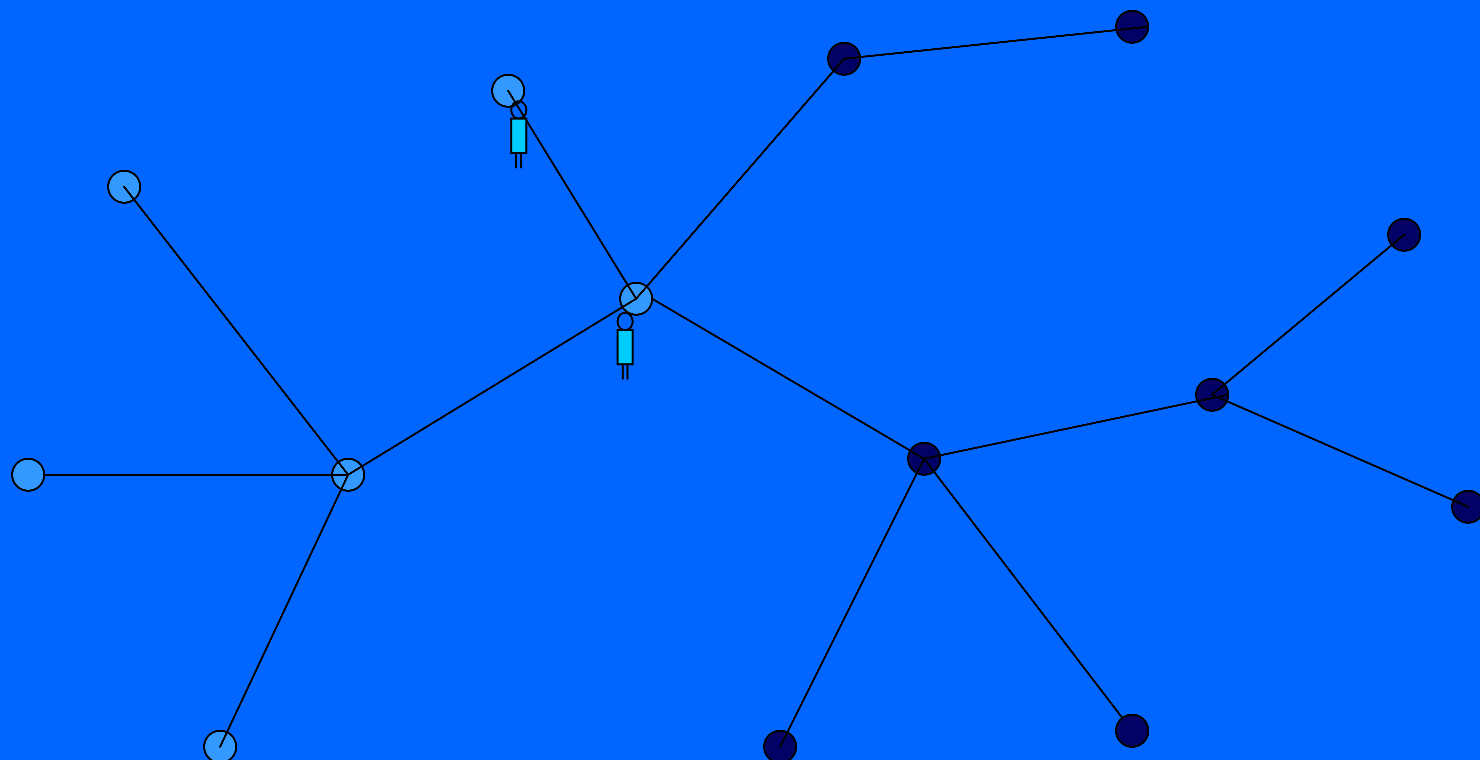


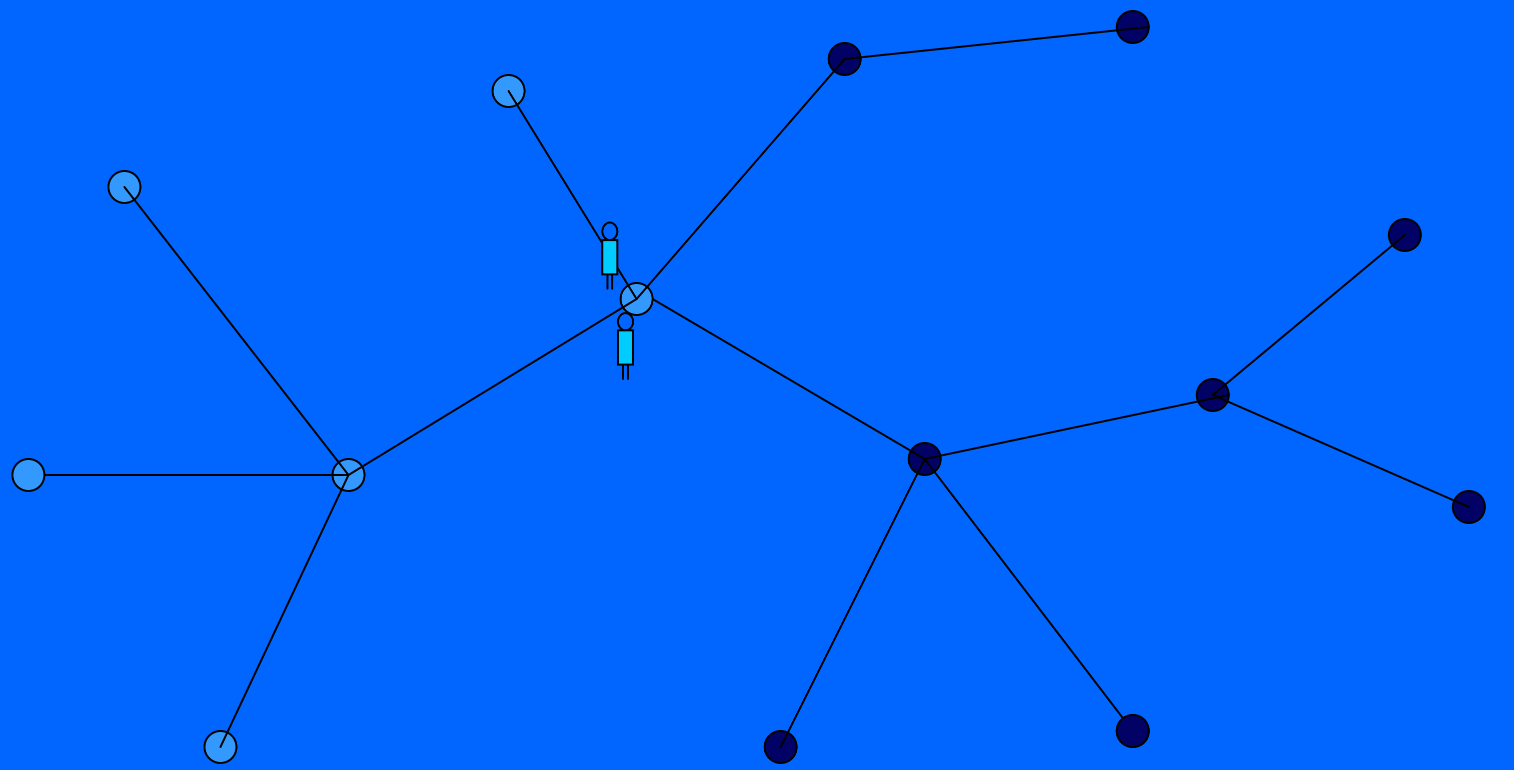


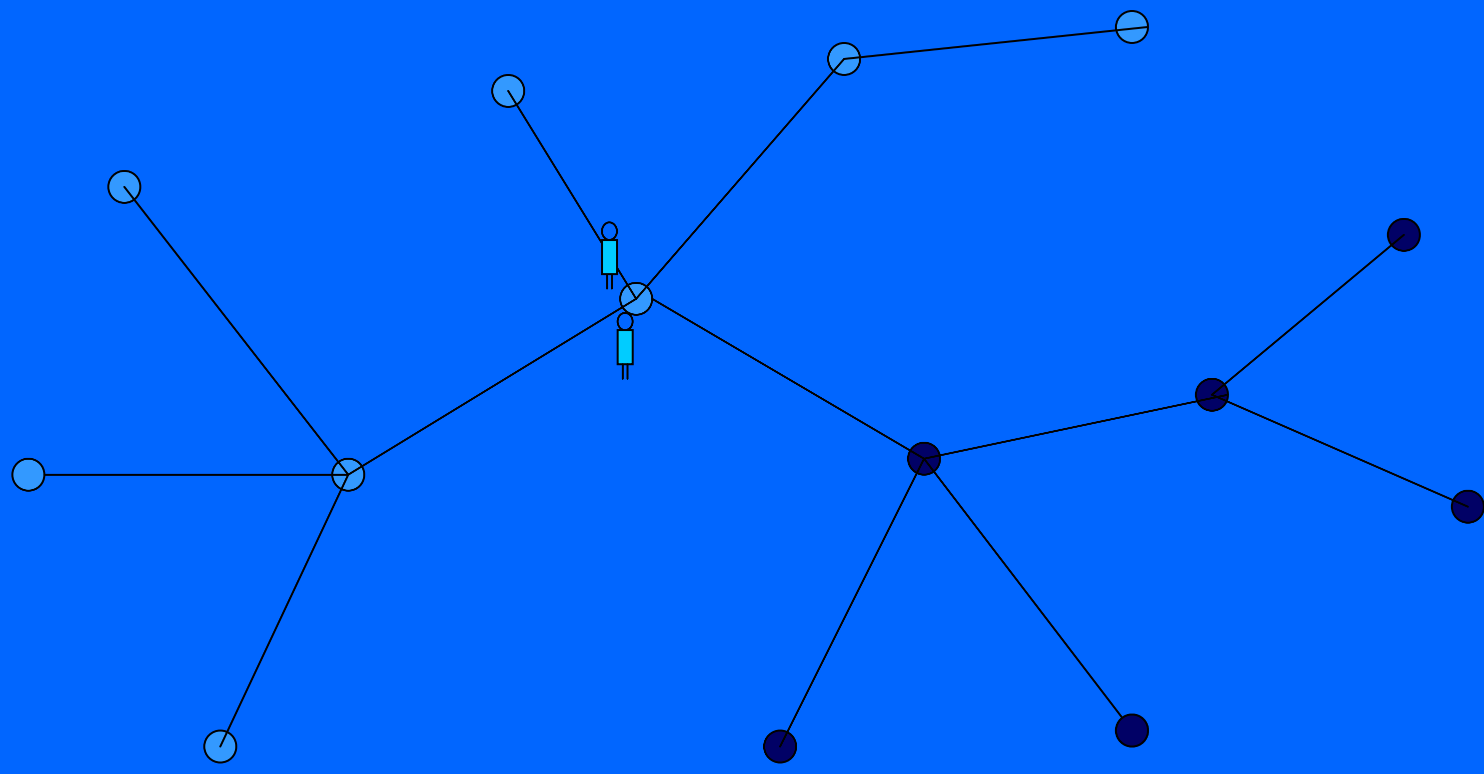


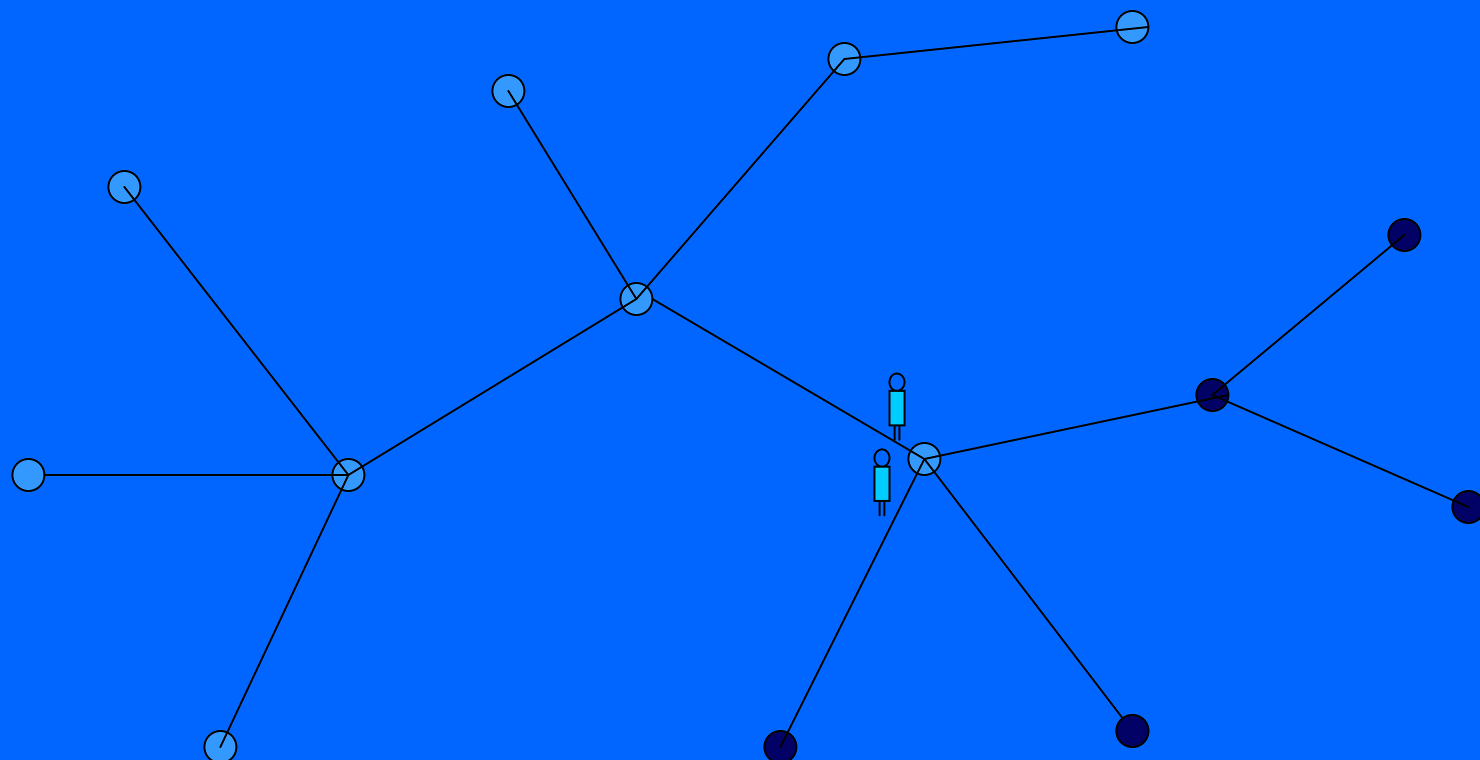


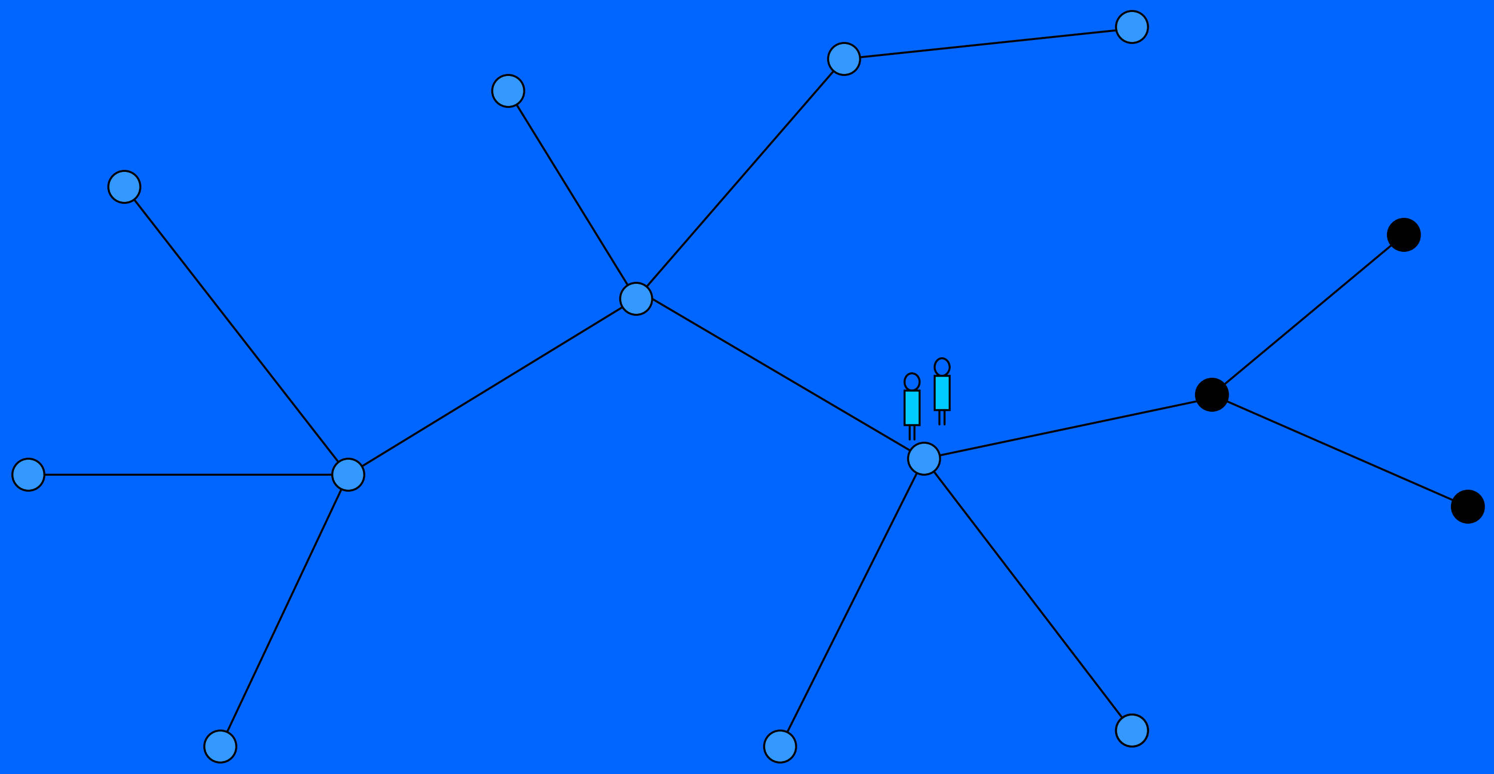


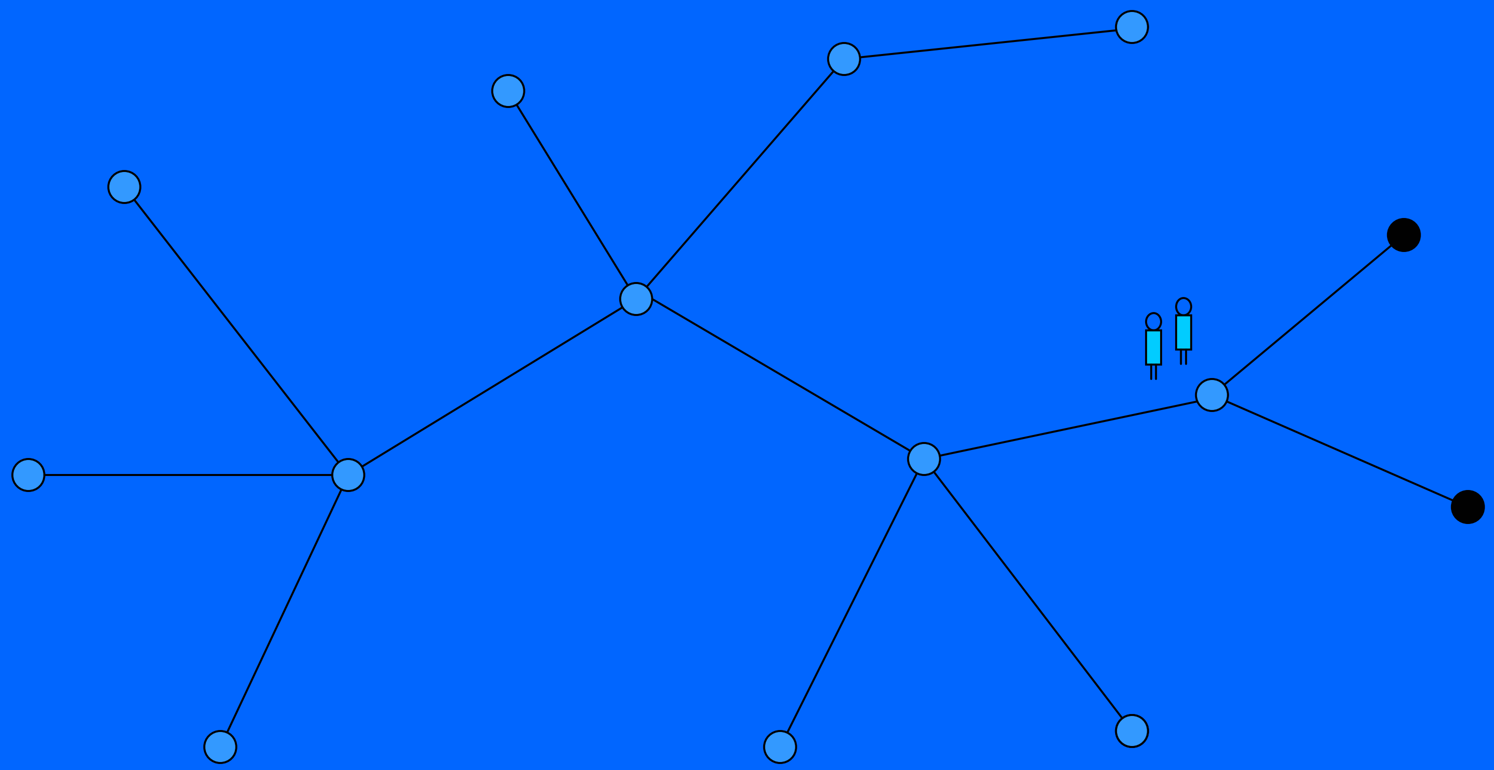


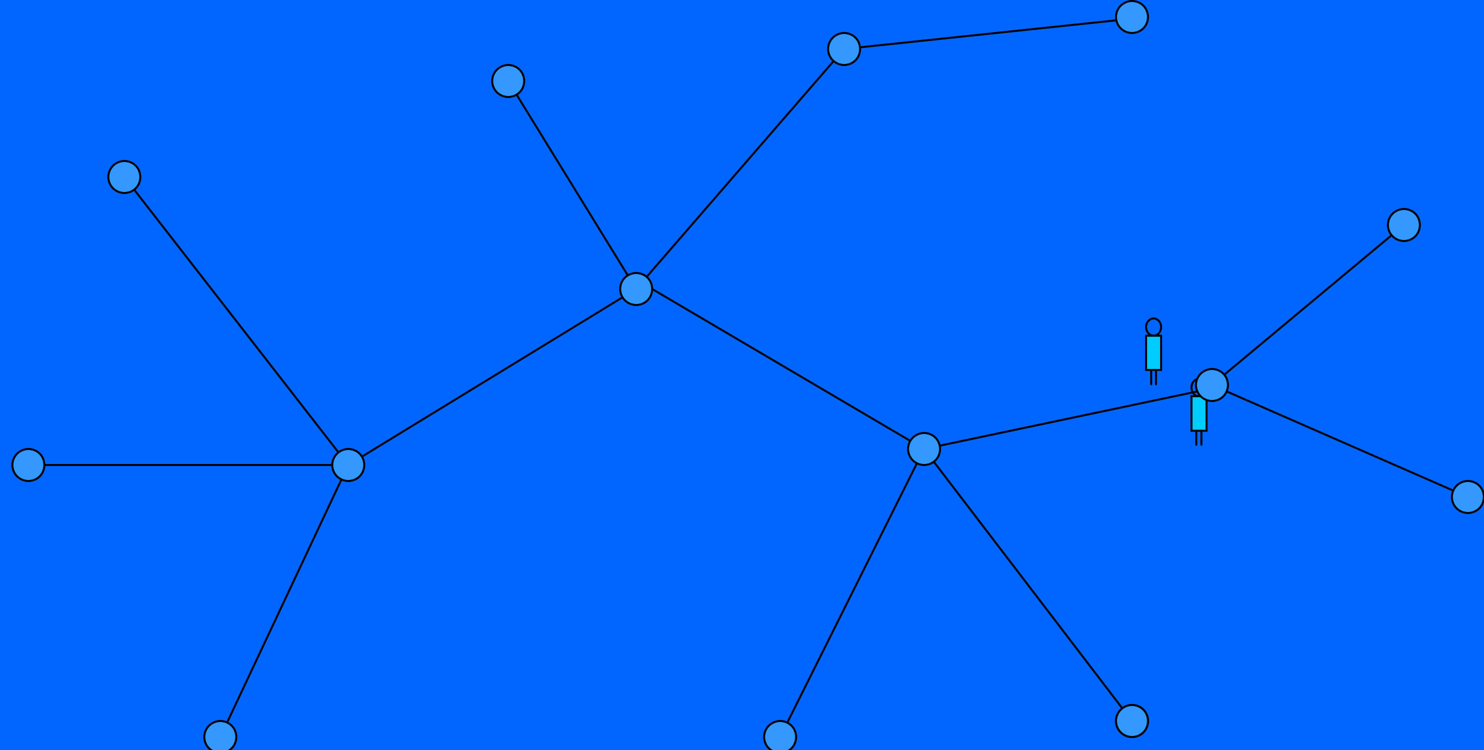




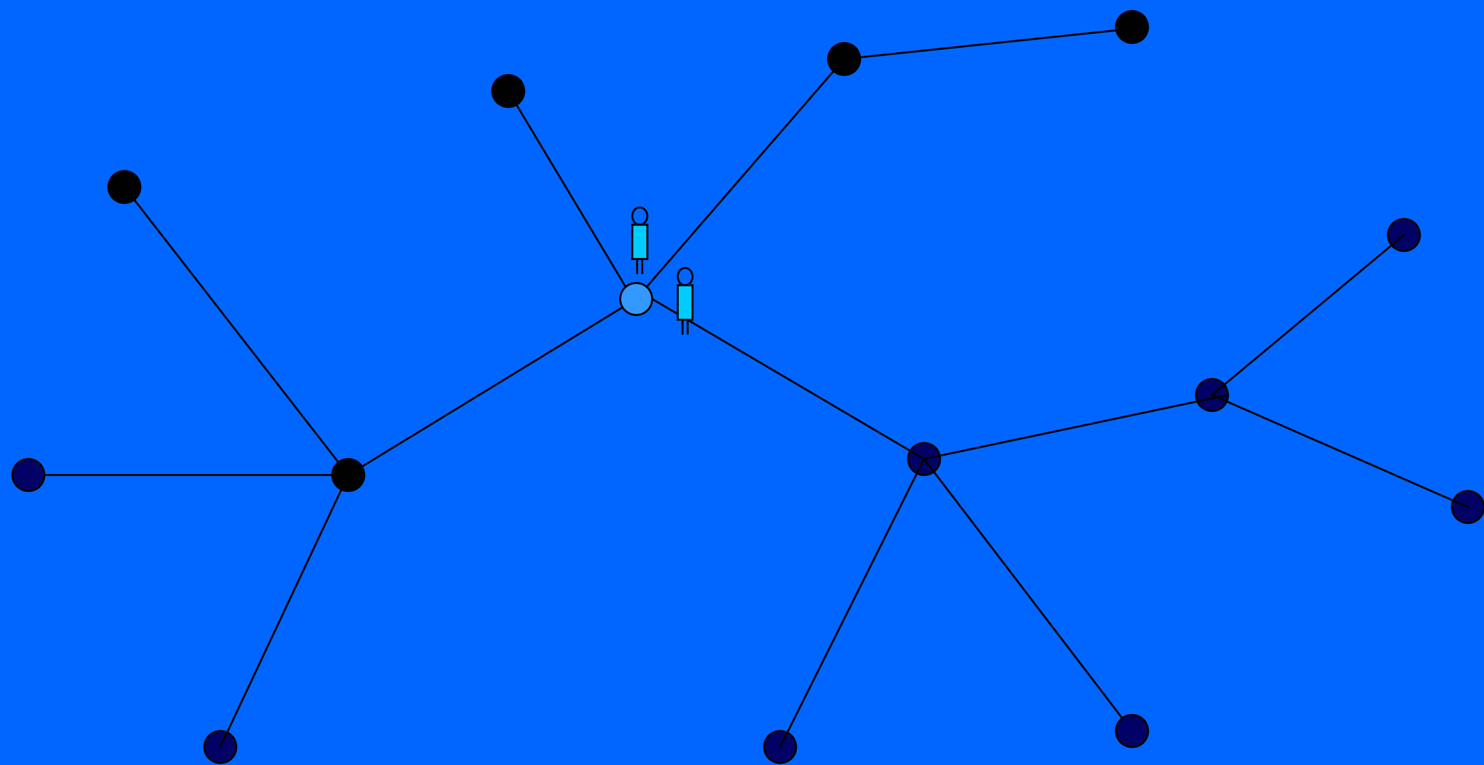




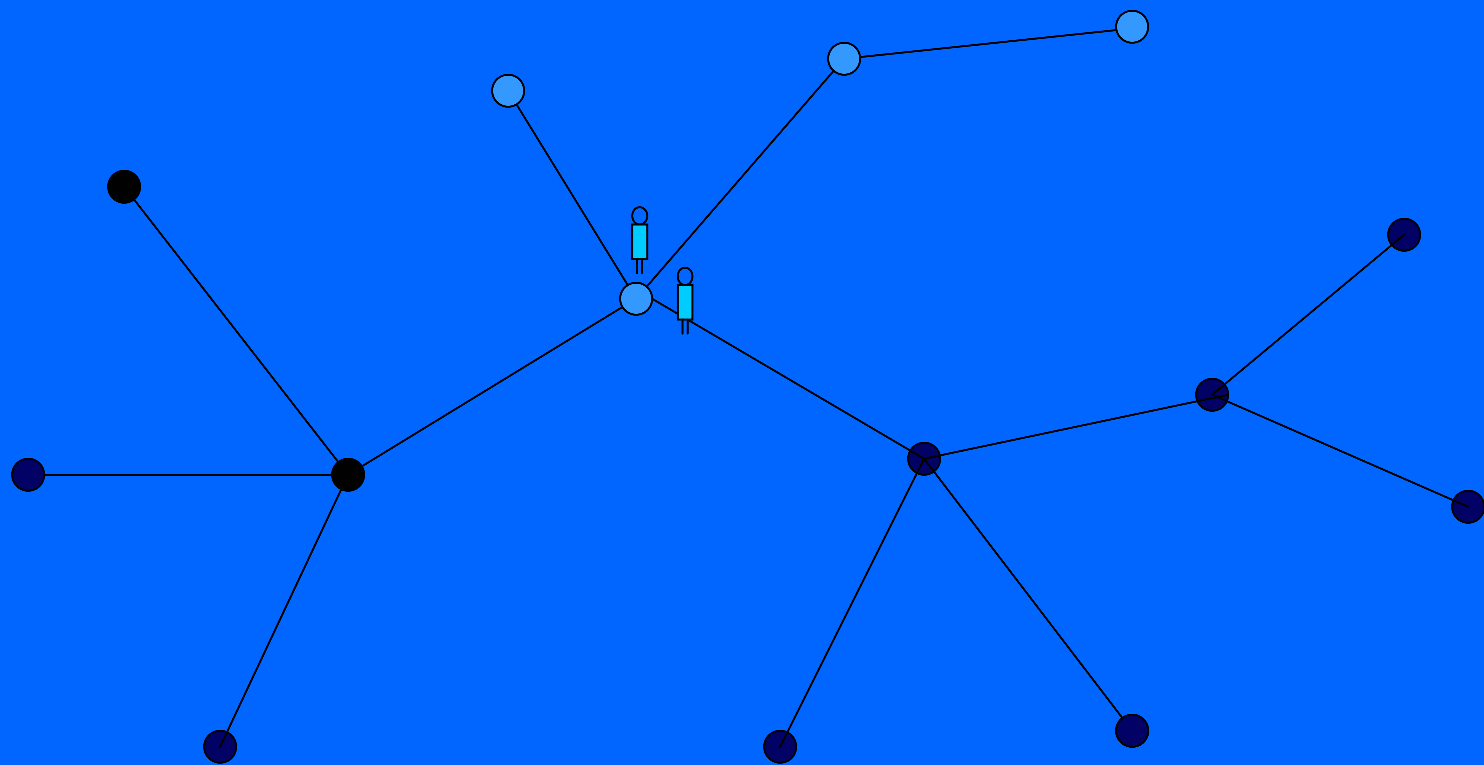




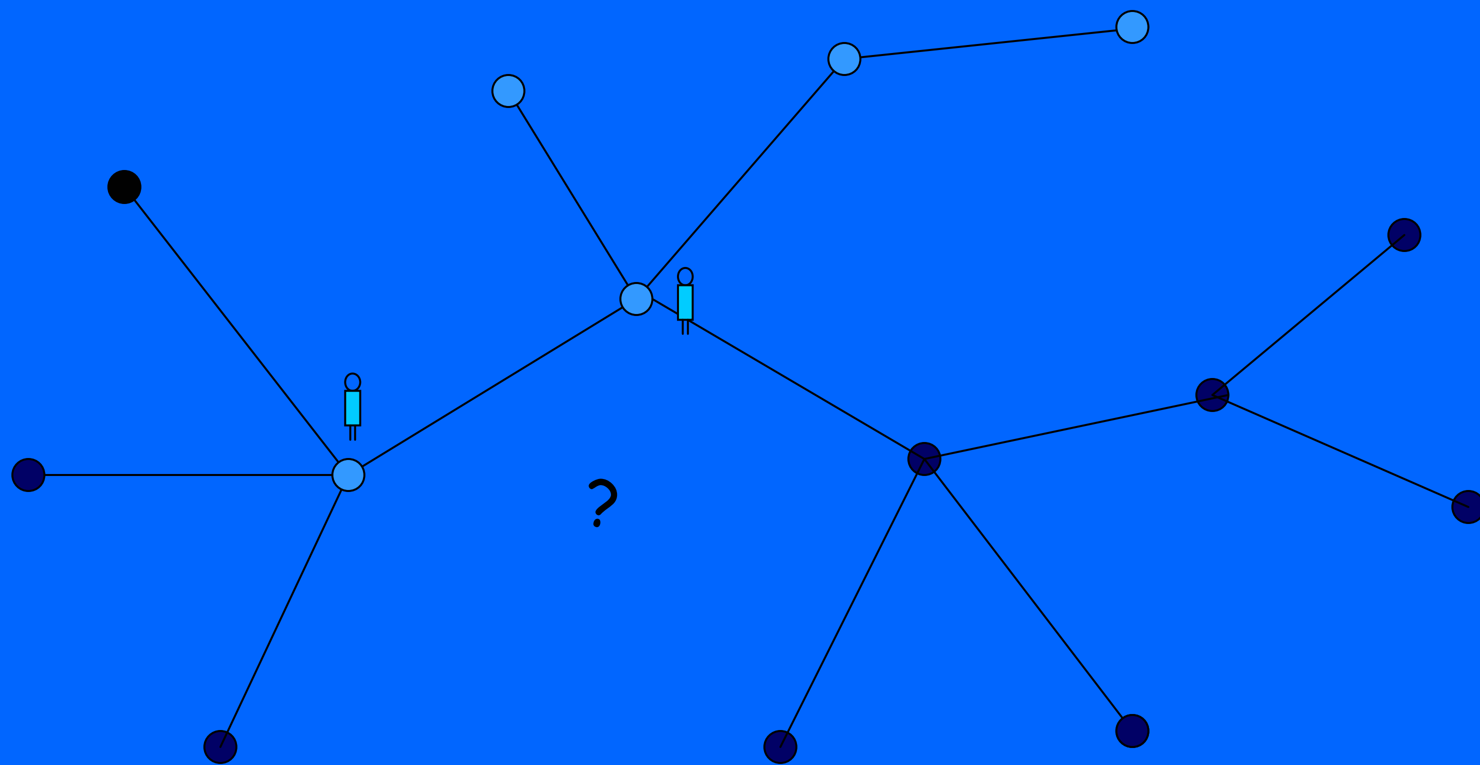
From a different starting point



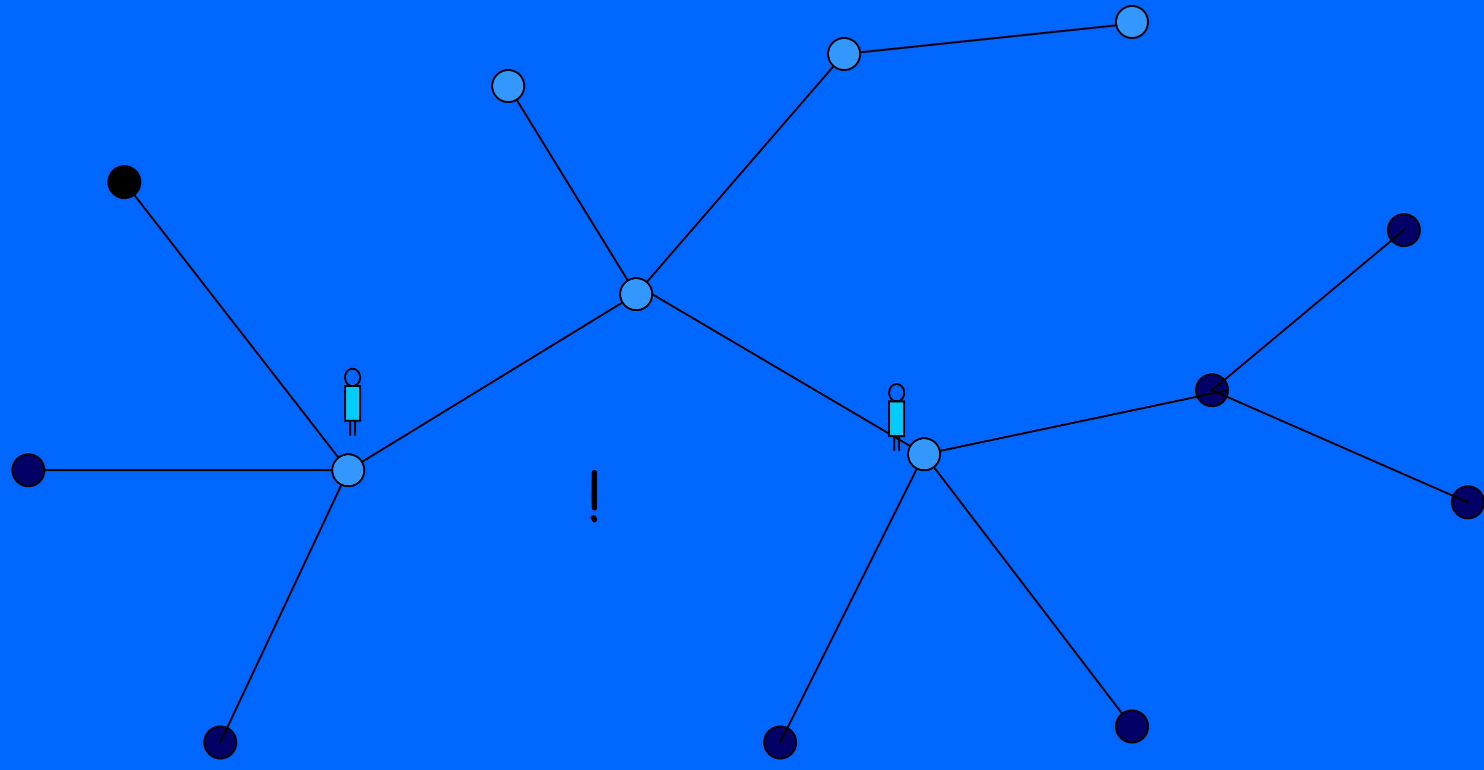
From a different starting point



From a different starting point

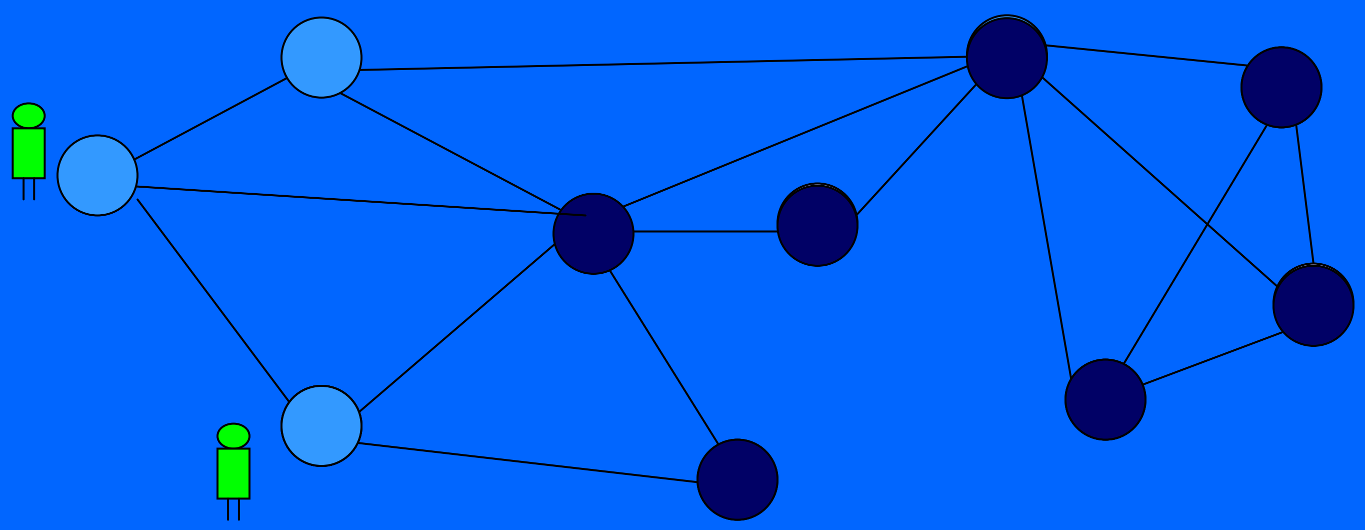


From a different starting point **two agents are not sufficient**



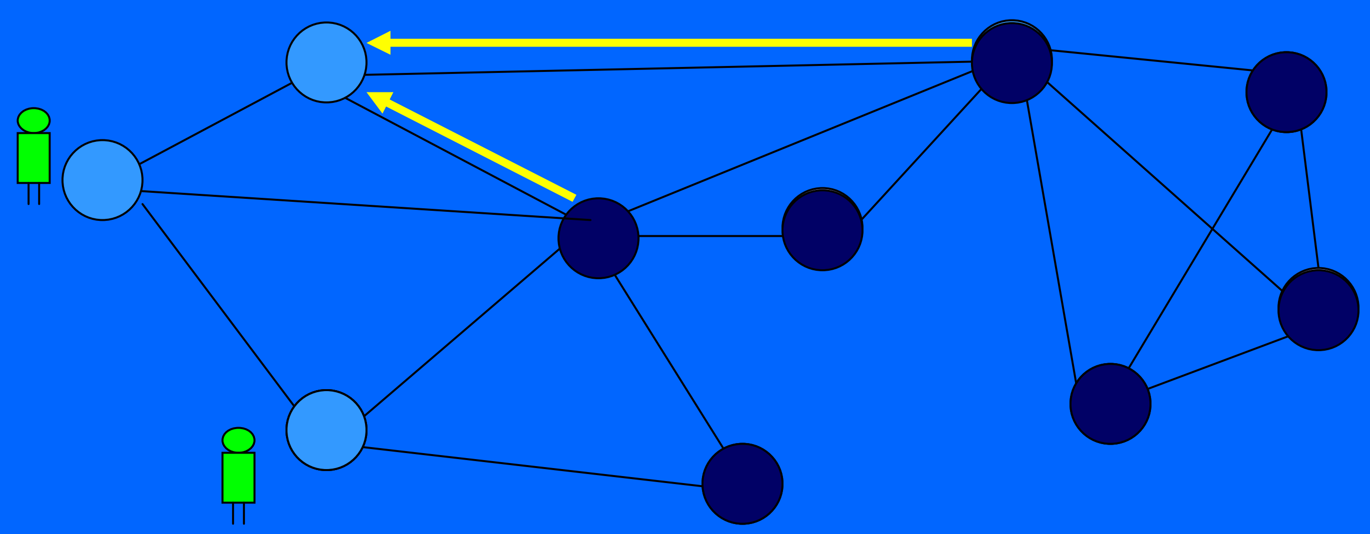
RECONTAMINATION BY MAJORITY

A clean but unguarded site will be recontaminated if a (strong) majority of its neighbours are contaminated



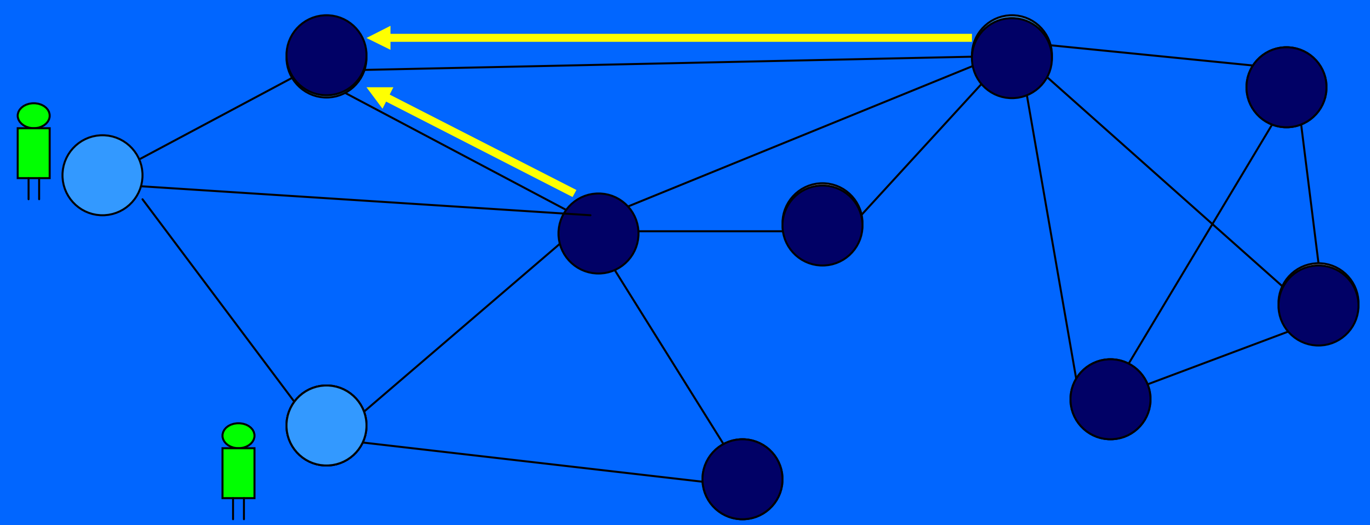
RECONTAMINATION BY MAJORITY

A clean but unguarded site will be recontaminated if a (strong) majority of its neighbours are contaminated



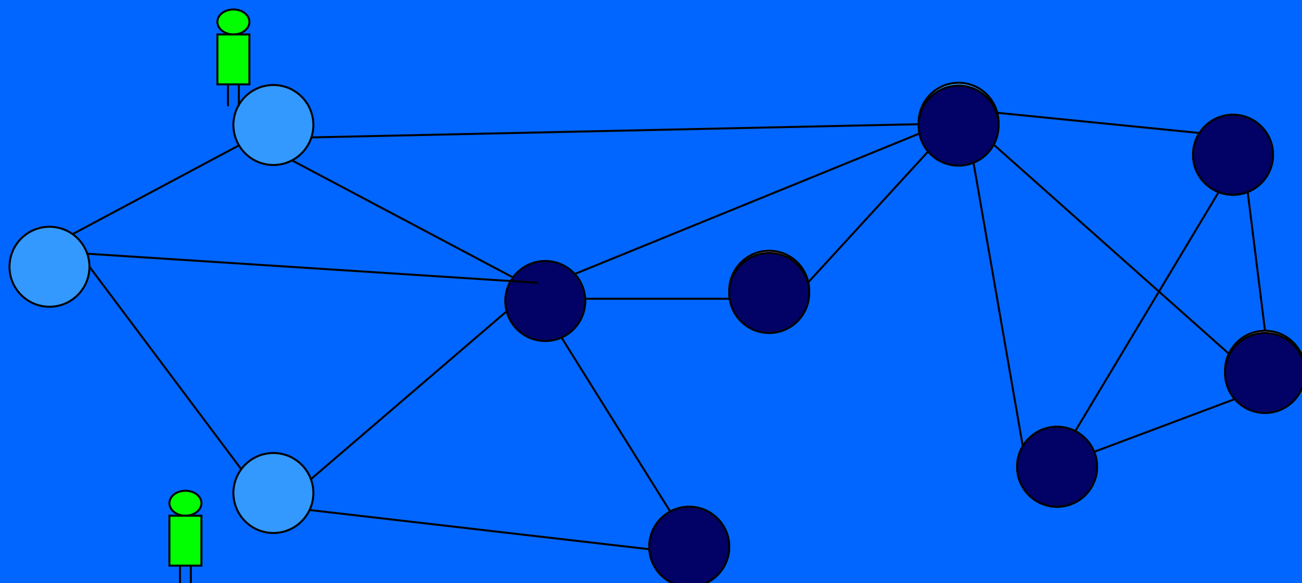
RECONTAMINATION BY MAJORITY

A clean but unguarded site will be recontaminated if a (strong) majority of its neighbours are contaminated



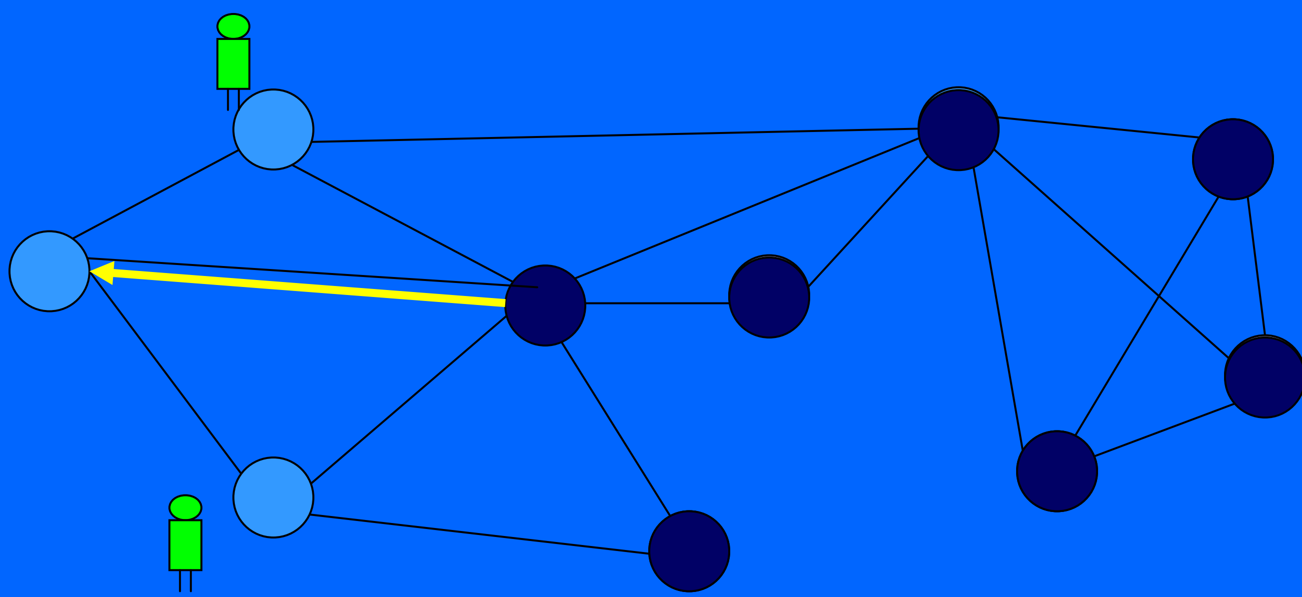
RECONTAMINATION
BY
MAJORITY

A clean but unguarded site will be recontaminated if a (strong) majority of its neighbours are contaminated



RECONTAMINATION
BY
MAJORITY

A clean but unguarded site will be recontaminated if a (strong) majority of its neighbours are contaminated

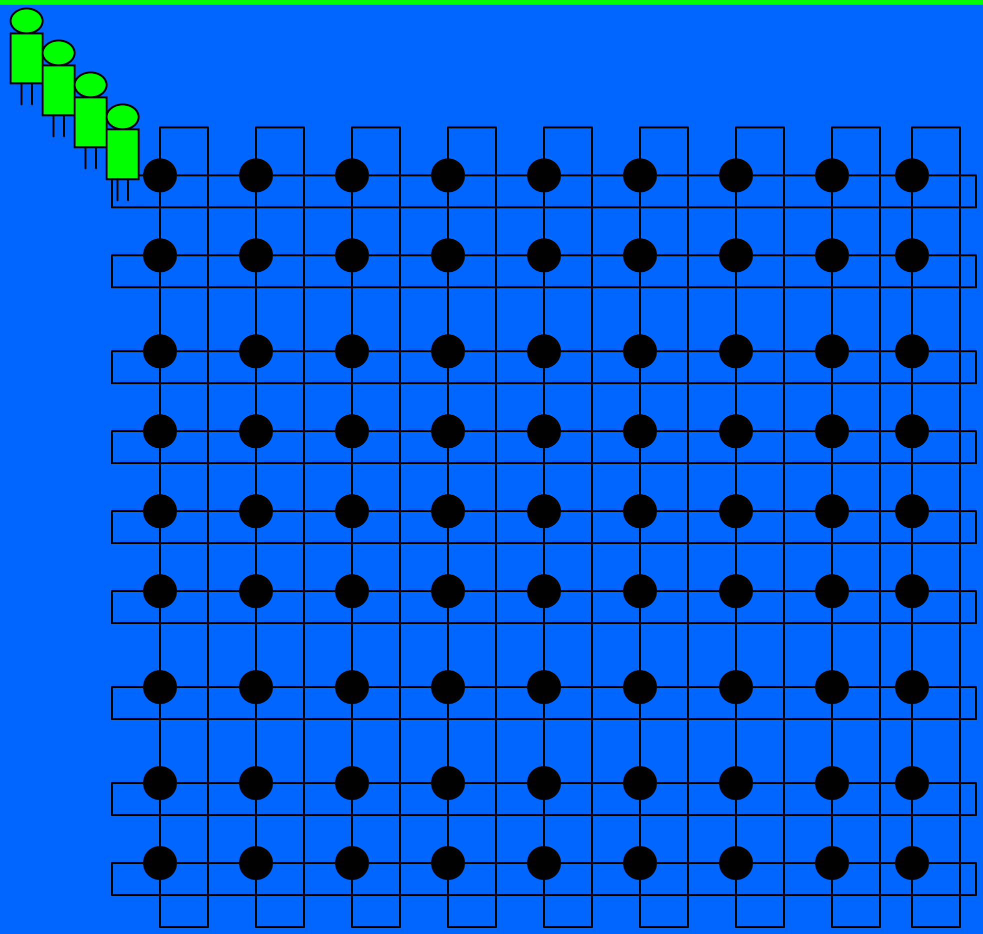


RECONTAMINATION
BY
MAJORITY

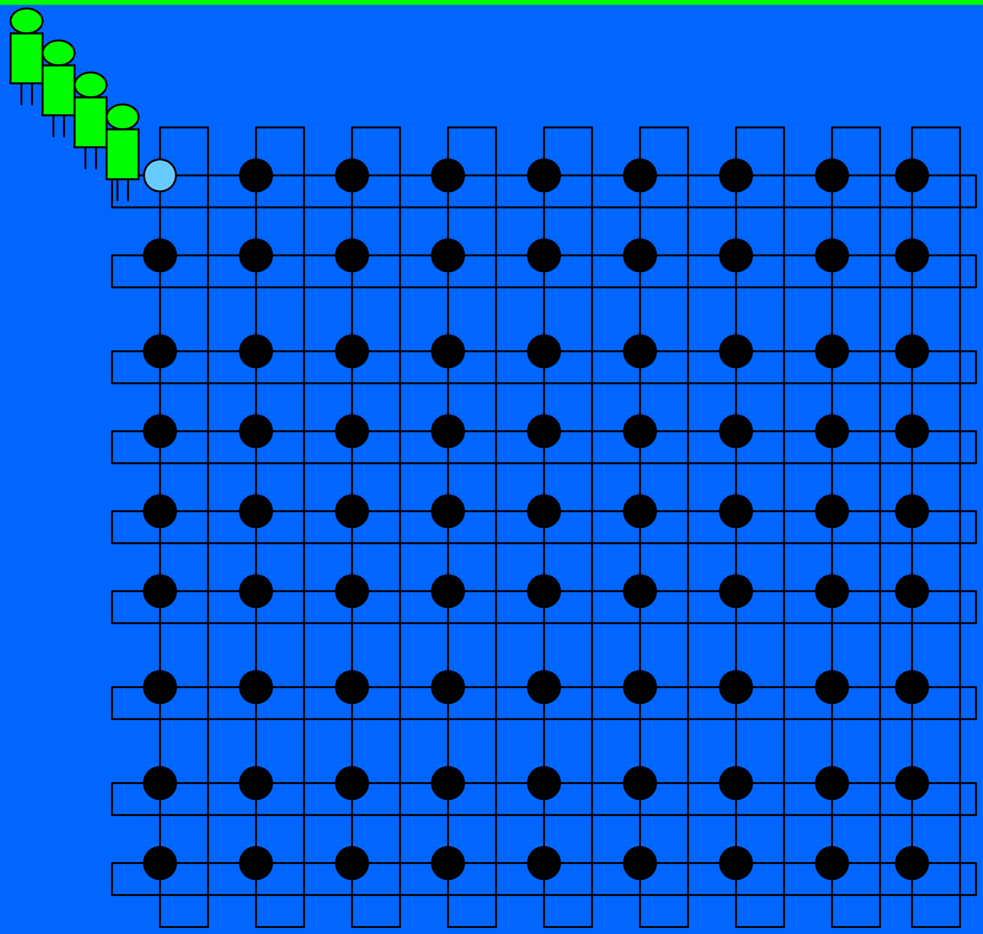
Very few results exist and only for
Toroidal Mesh (2-dimensional torus)
and k-Trees

[Int. J. of Foundation of Computer Science: Luccio et al 2006]

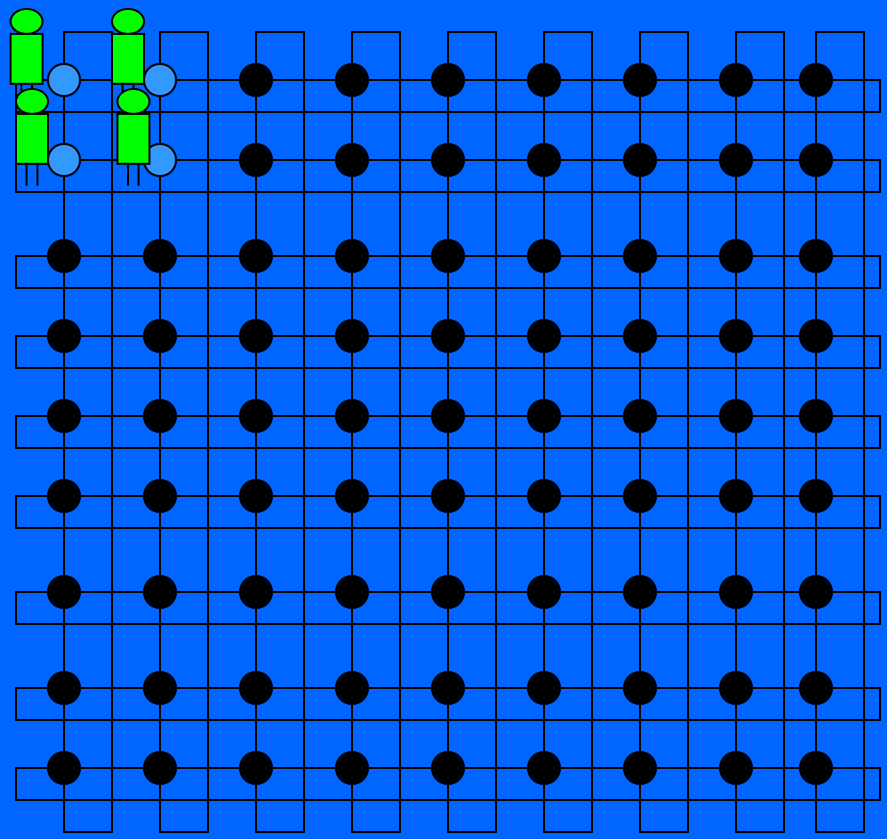
Toroidal Mesh (2-dimensional torus)



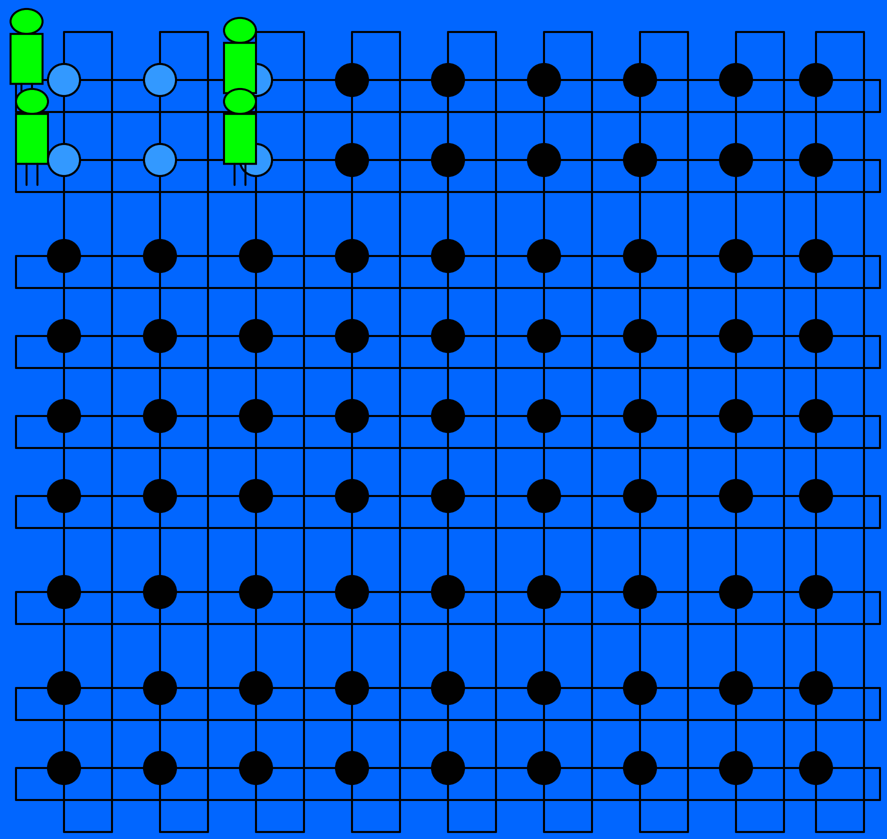
Toroidal Mesh (2-dimensional torus)



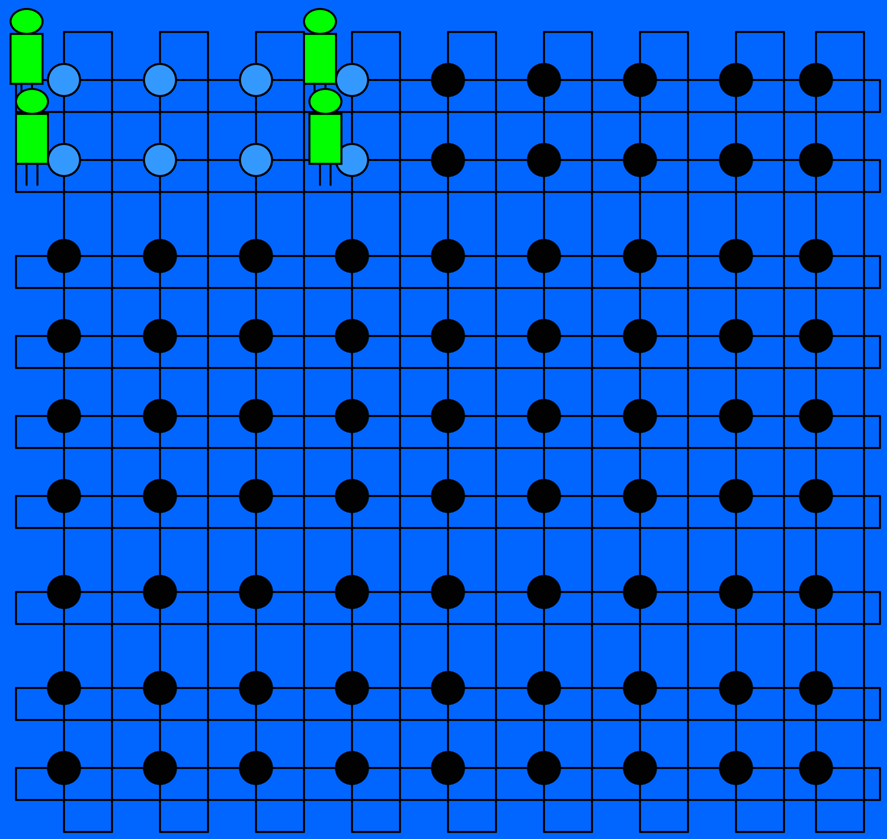
Toroidal Mesh (2-dimensional torus)



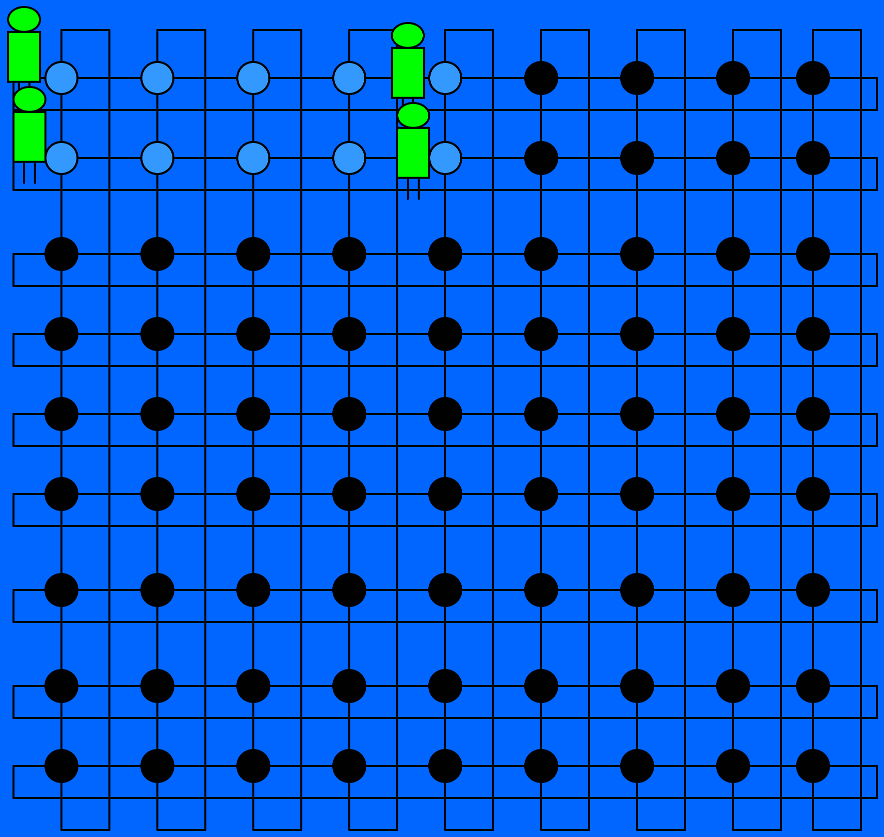
Toroidal Mesh (2-dimensional torus)



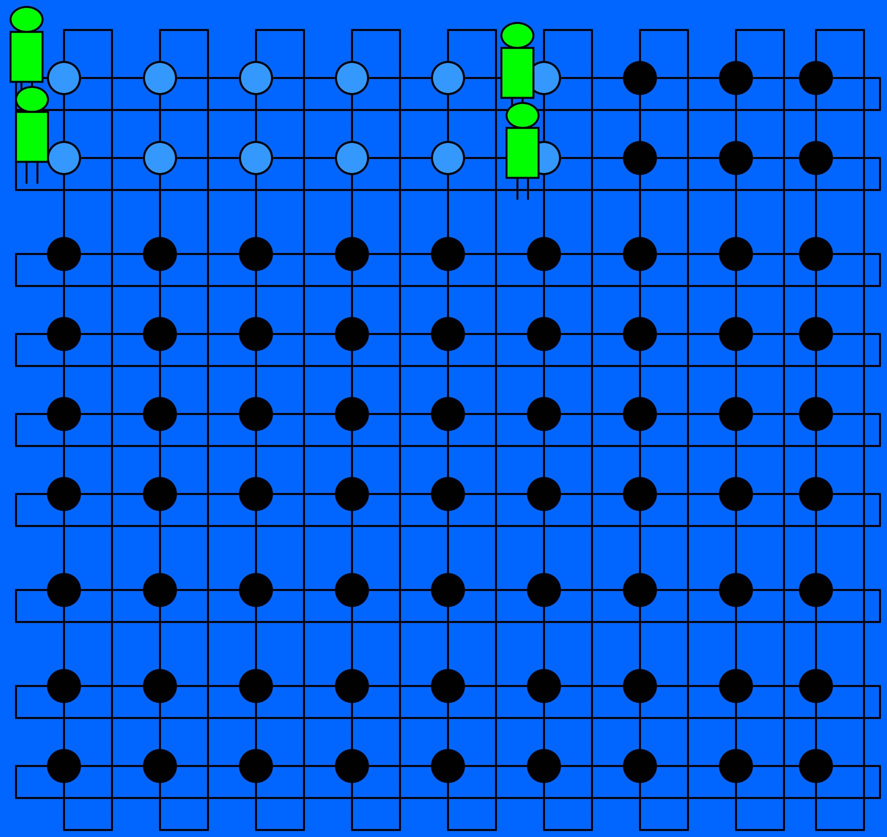
Toroidal Mesh (2-dimensional torus)



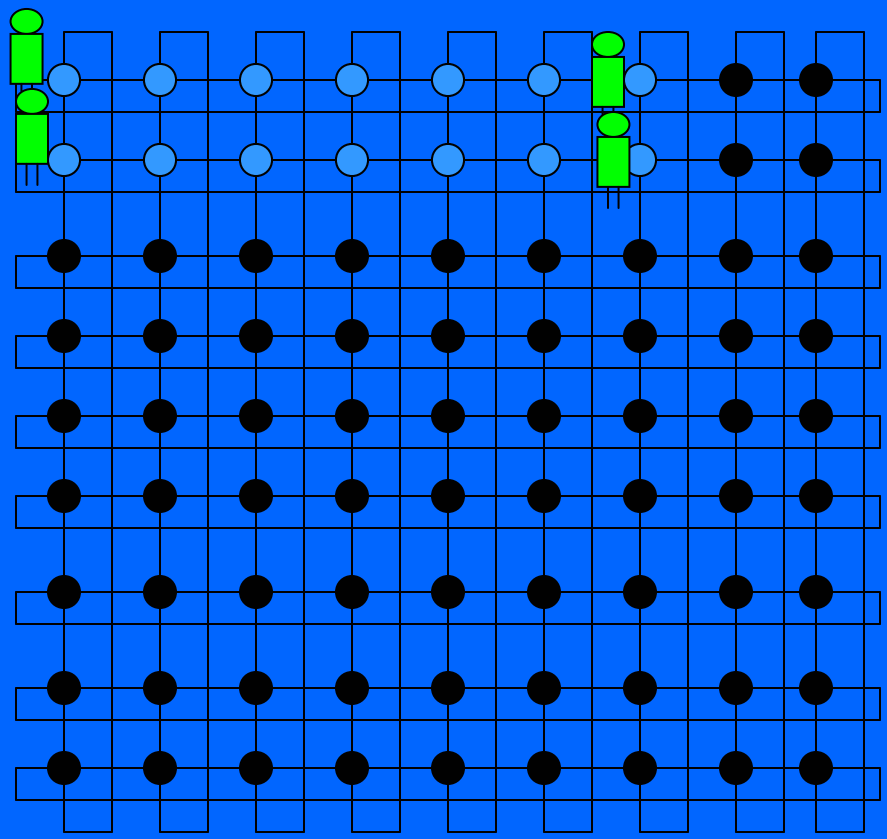
Toroidal Mesh (2-dimensional torus)



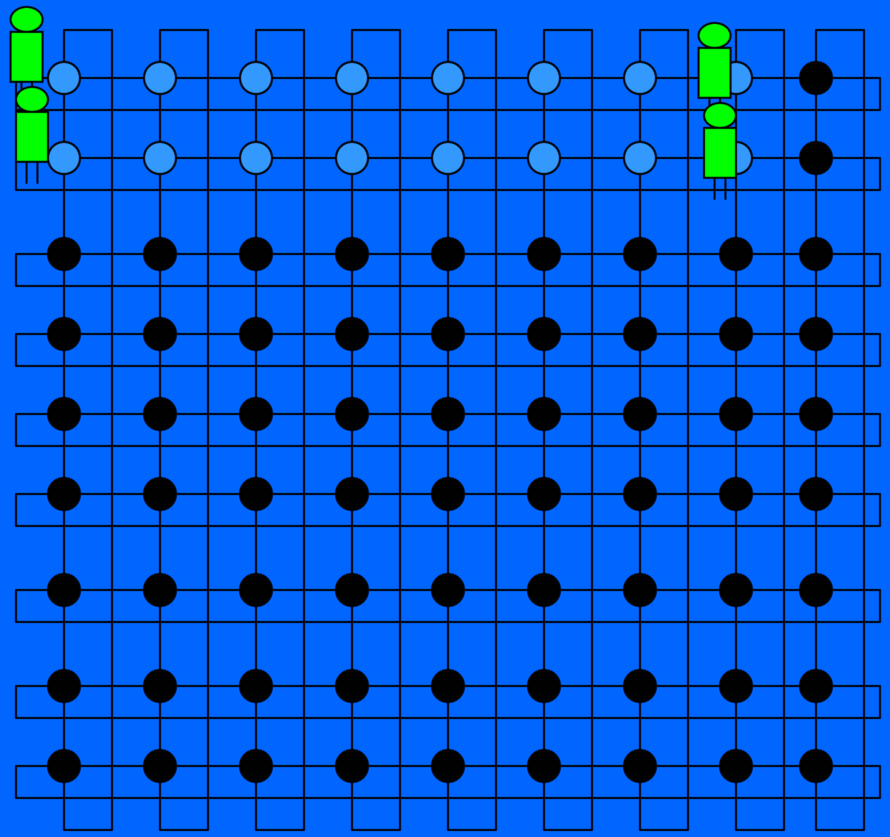
Toroidal Mesh (2-dimensional torus)



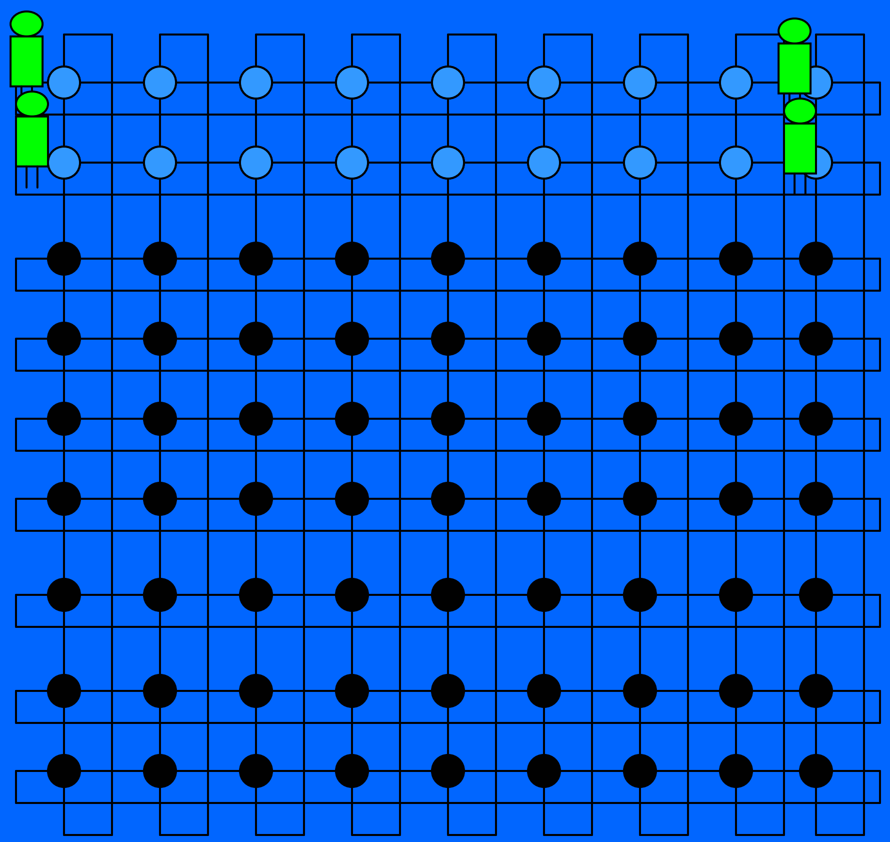
Toroidal Mesh (2-dimensional torus)



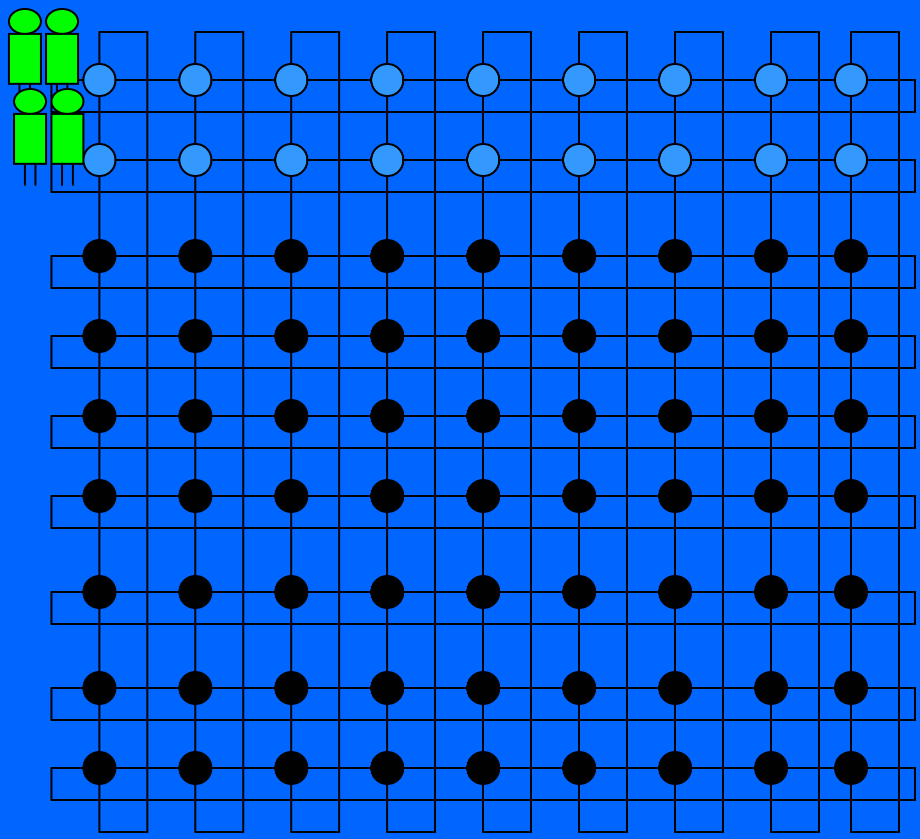
Toroidal Mesh (2-dimensional torus)



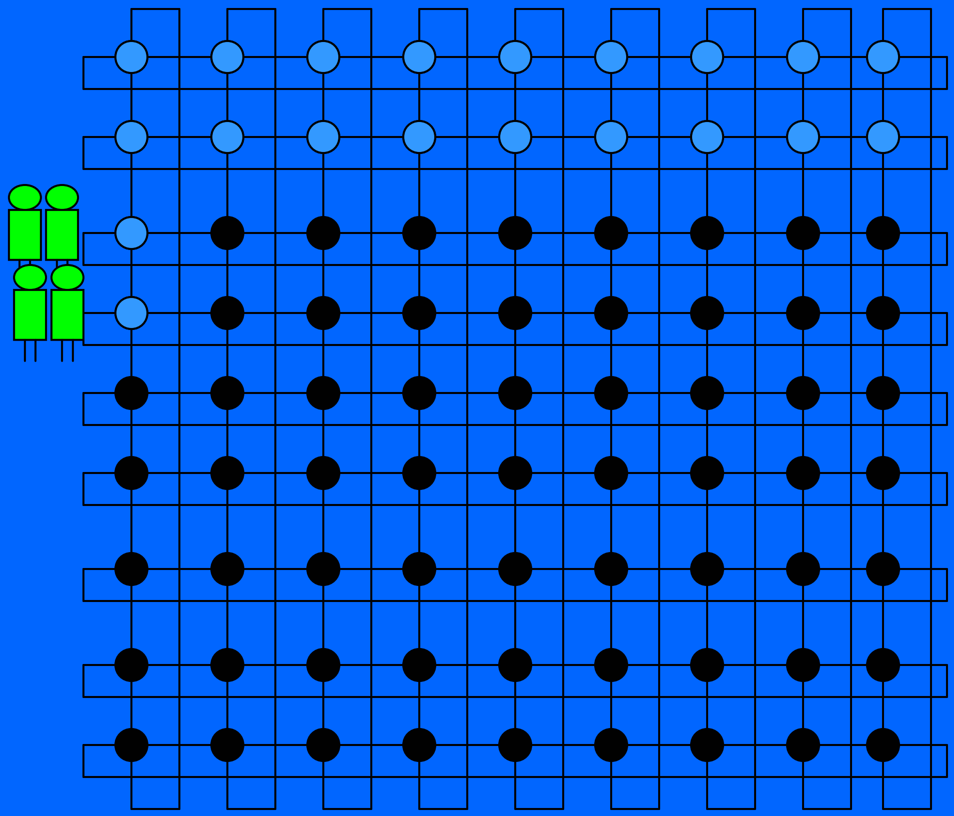
Toroidal Mesh (2-dimensional torus)



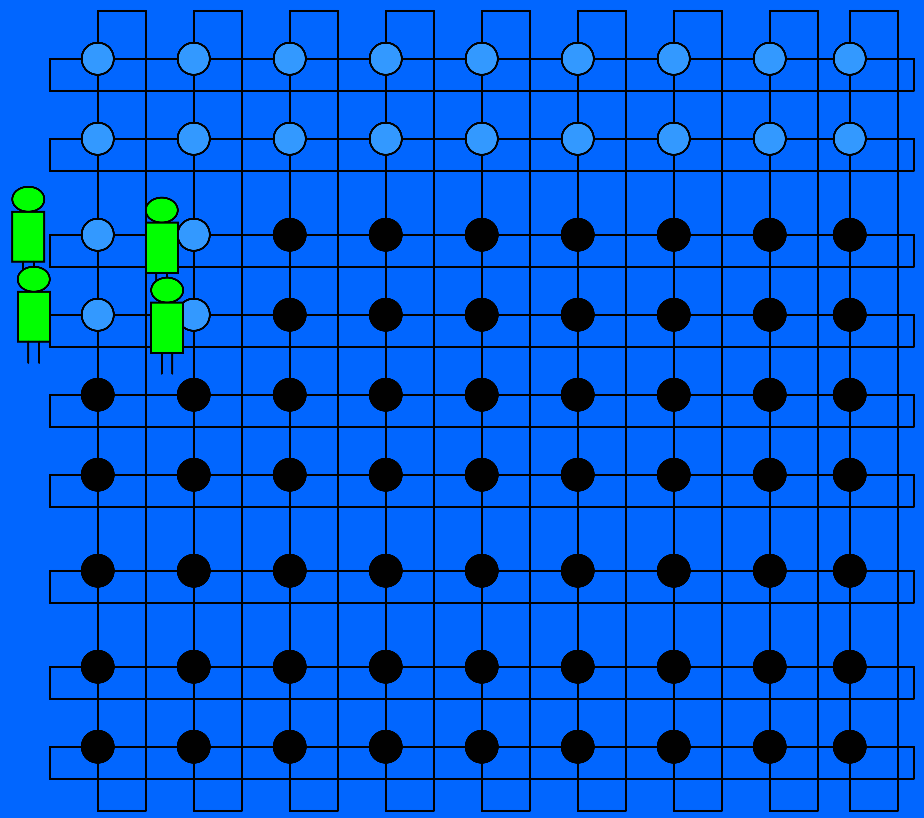
Toroidal Mesh (2-dimensional torus)



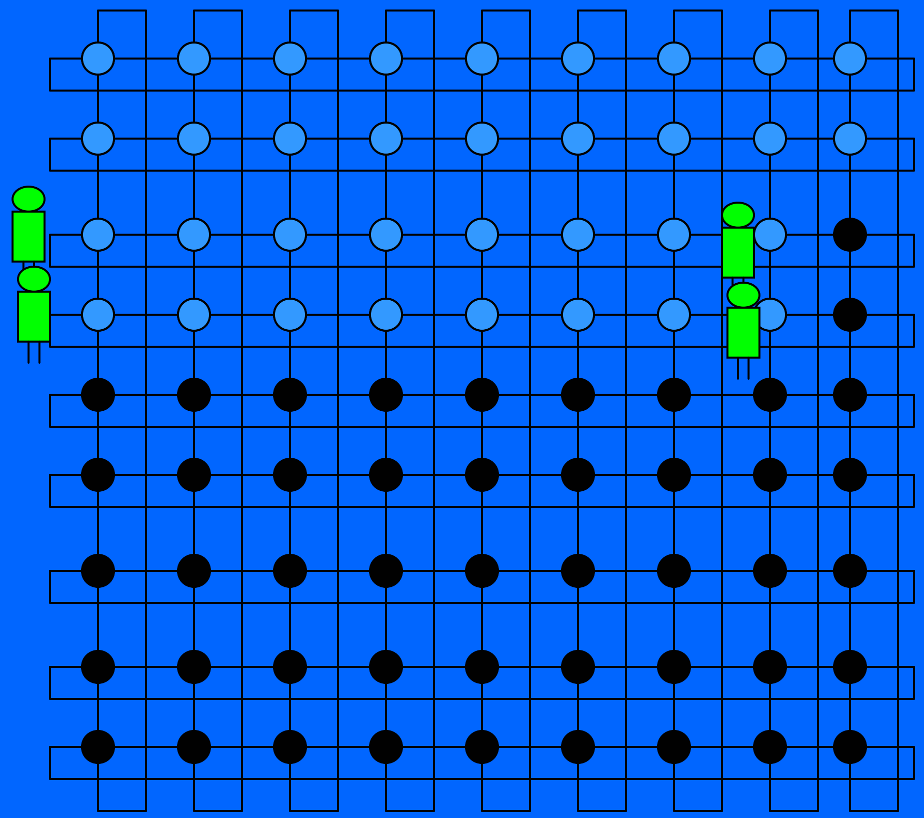
Toroidal Mesh (2-dimensional torus)



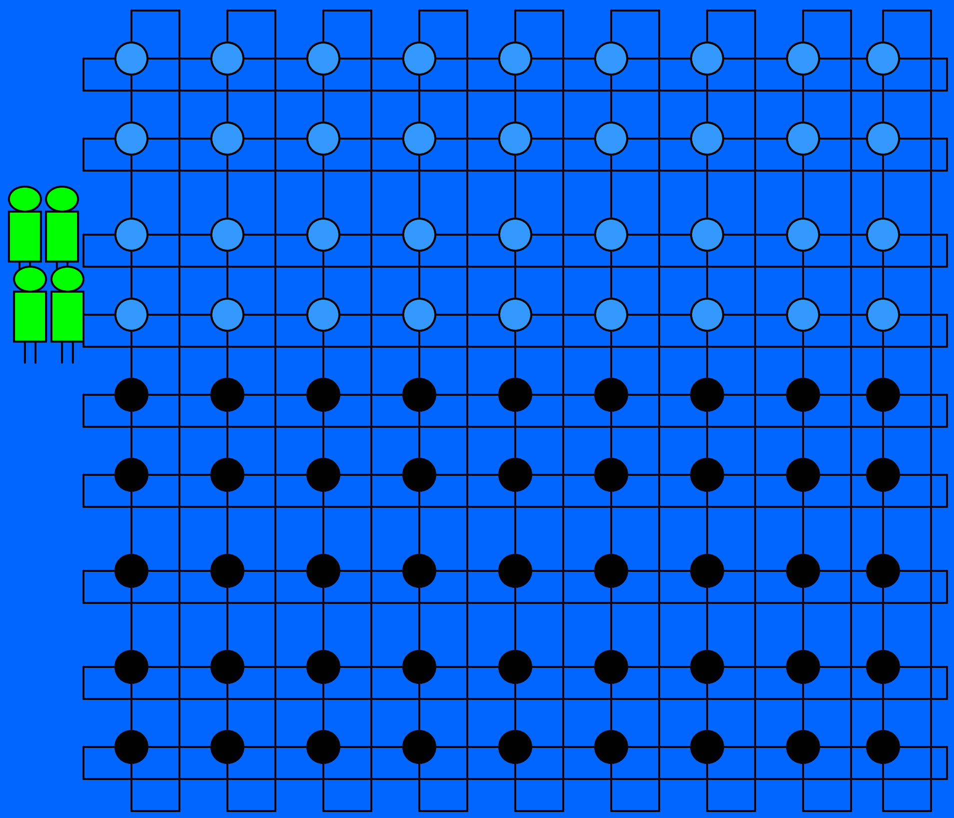
Toroidal Mesh (2-dimensional torus)



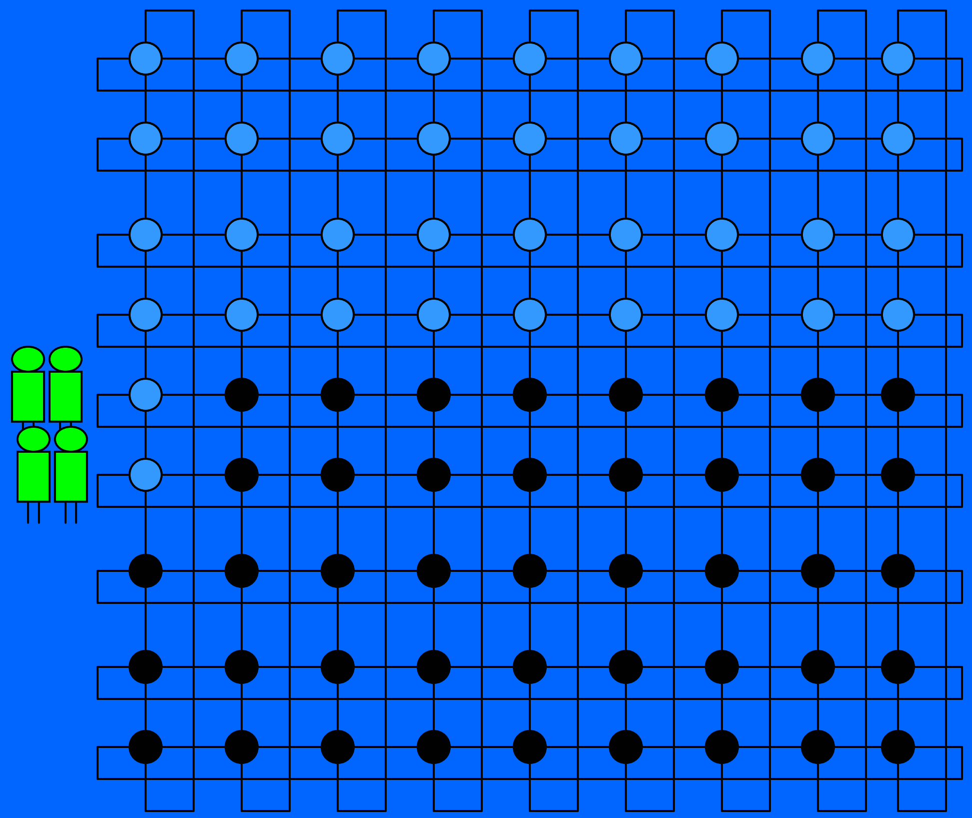
Toroidal Mesh (2-dimensional torus)



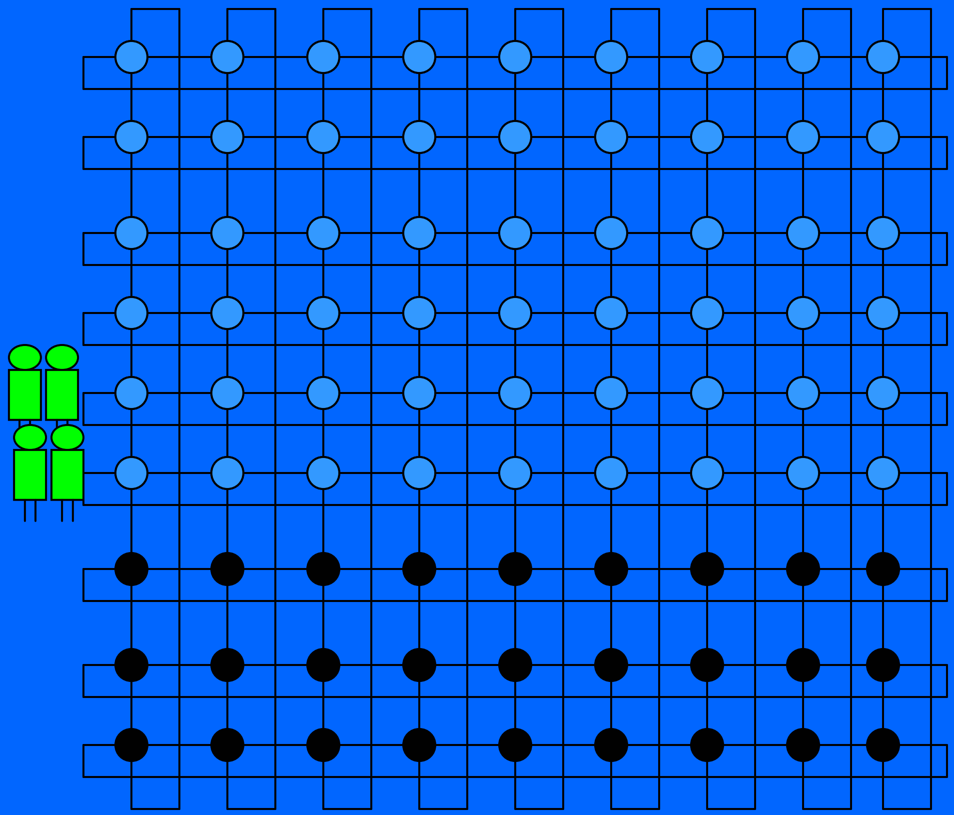
Toroidal Mesh (2-dimensional torus)



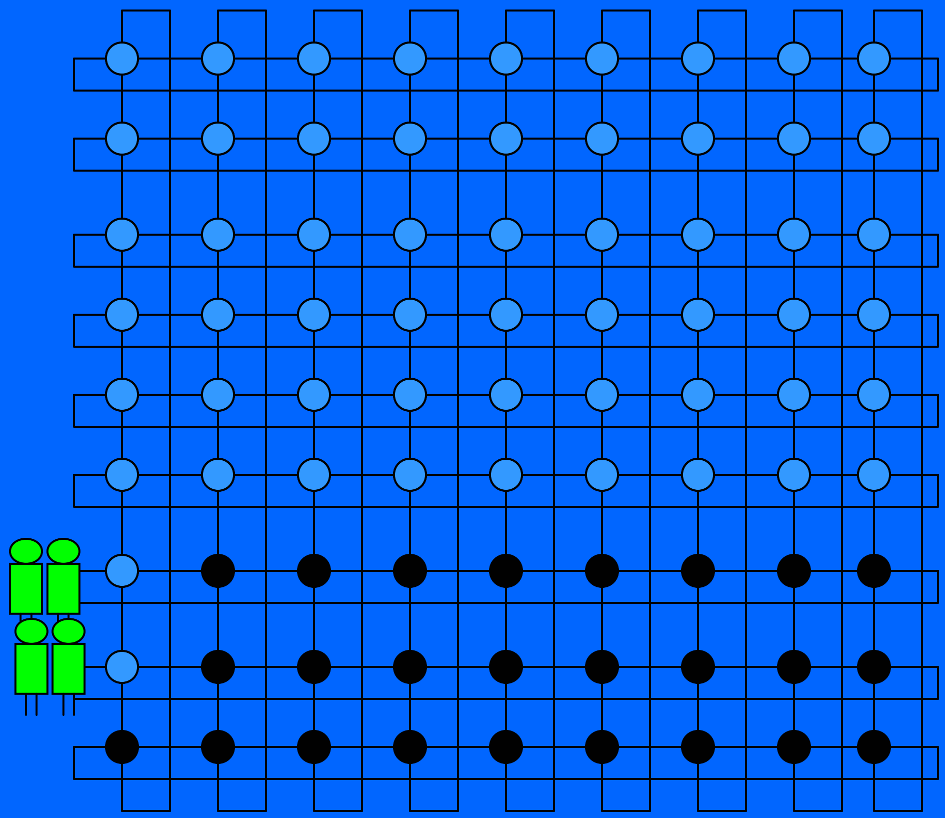
Toroidal Mesh (2-dimensional torus)



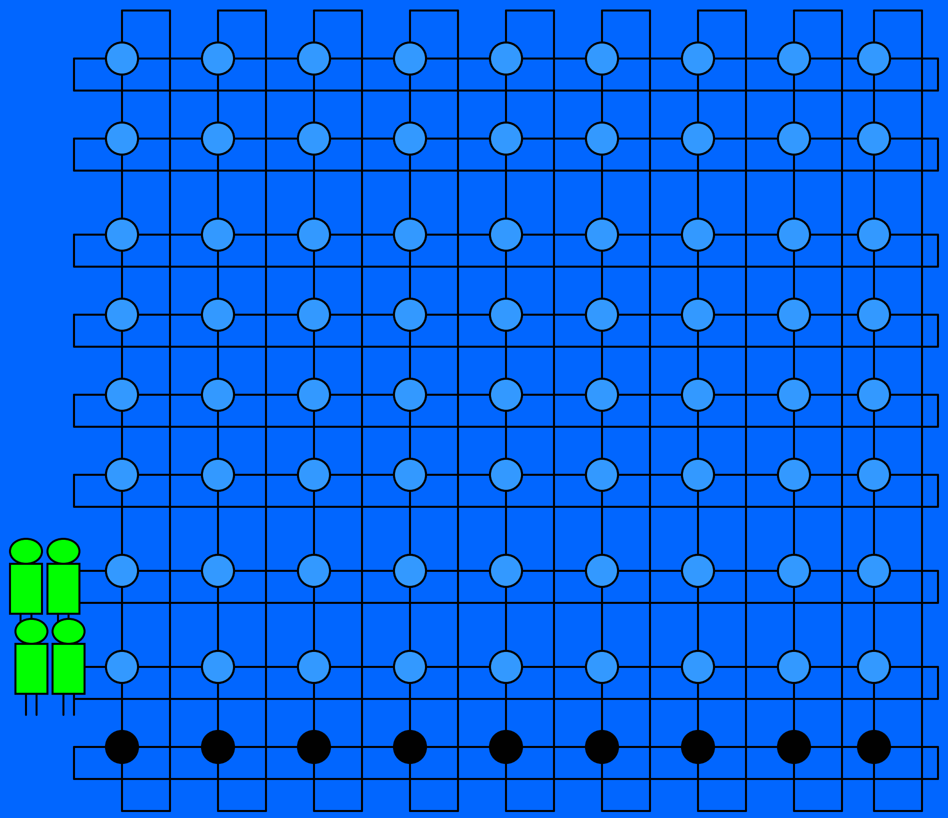
Toroidal Mesh (2-dimensional torus)



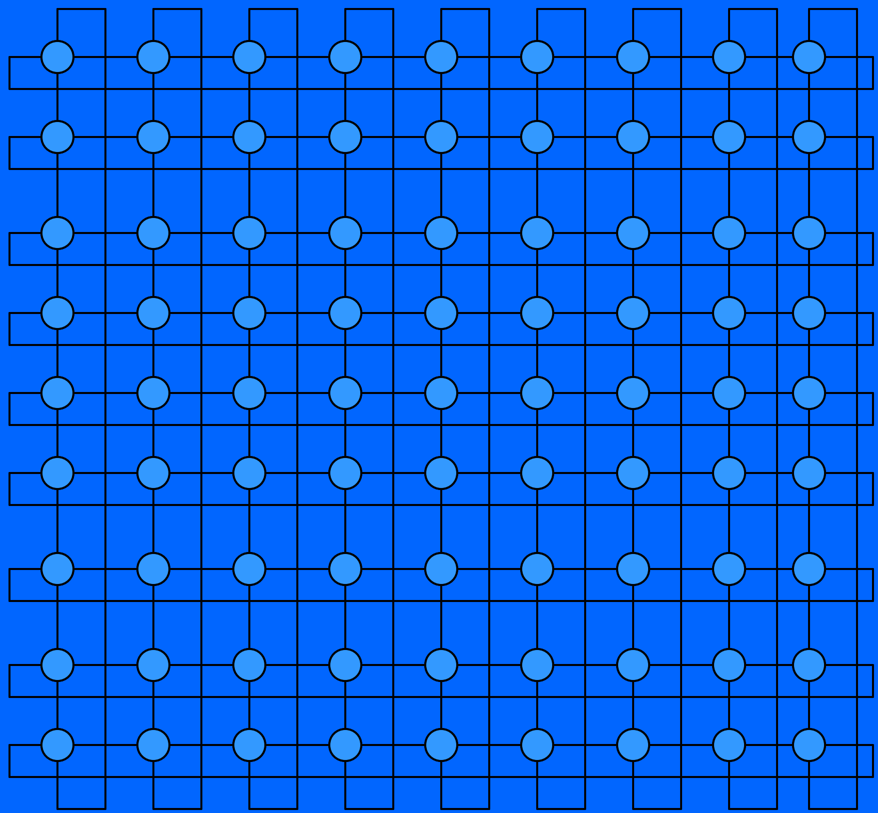
Toroidal Mesh (2-dimensional torus)



Toroidal Mesh (2-dimensional torus)



Toroidal Mesh (2-dimensional torus)



(not the optimal algorithm)

Observations

In Meshes

Size of Team when **1 contaminated neighbour suffices** for recontamination: **O(m+n)**

Size of Team when **majority of contaminated neighbours is necessary** for recontamination: **O(1)**

Size of Dynamos: O(m+n)

In Trees

The recontamination rule does not seem to matter much in the W.C. **O(log n)**

Research Directions

Dynamos:

- Reversible, non-monotone models
- Opposite problem: What are the self-healing configurations in various topologies ?
- Partial disruption (*e.g.*, guarantee that the majority of sites are operational)

…

Research Directions

Decontamination

- ALMOST NOTHING IS KNOWN

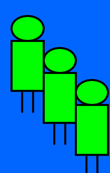
Research Directions

Randomized solutions ?

Other recontamination rules ?
(unanimity, other thresholds ….)

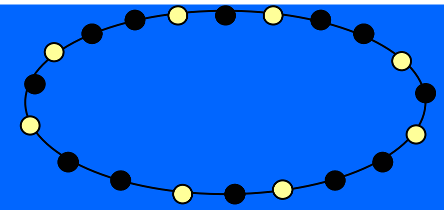
Find minimum team for a given initial configuration

THANK YOU



Observation

The Ring is a circular 1-dim CA



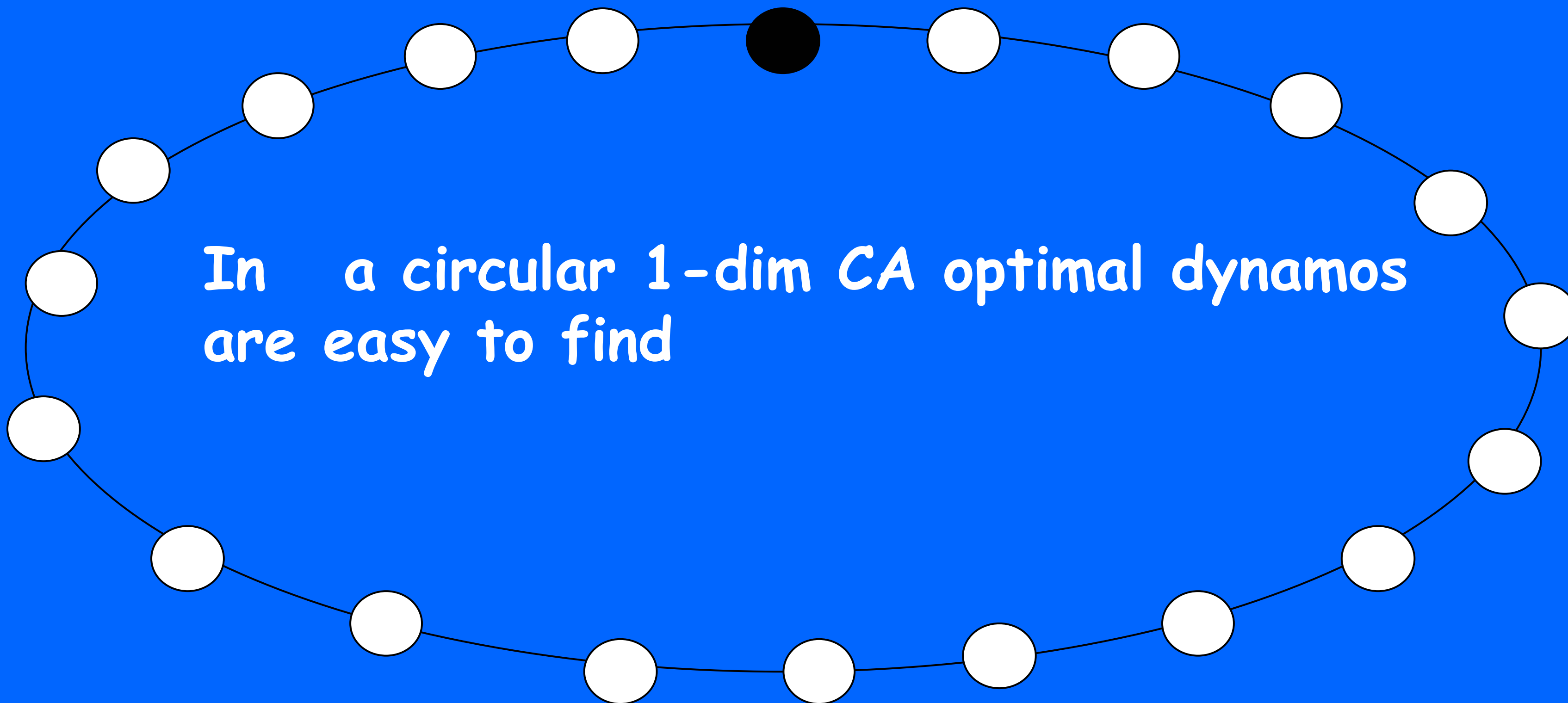
Simple Majority

	Irreversible (non-self)	Reversible (non-self)	Reversible (self)	Irreversible (self)
000	0	0	0	0
001	1	1	0	0
010	1	0	0	1
011	1	1	1	1
100	1	1	0	0
101	1	1	1	1
110	1	1	1	1
111	1	1	1	1
	255	251	233	237

Strong Majority

	Irreversible (non-self)	Reversible (non-self)	Reversible (self)	Irreversible (self)
000	0	0	0	0
001	0	0	0	0
010	1	0	0	1
011	1	0	1	1
100	0	0	0	0
101	1	1	1	1
110	1	0	1	1
111	1	1	1	1
	236	160	233	237

In a circular 1-dim CA optimal dynamos are easy to find

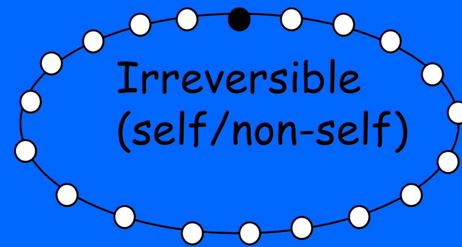


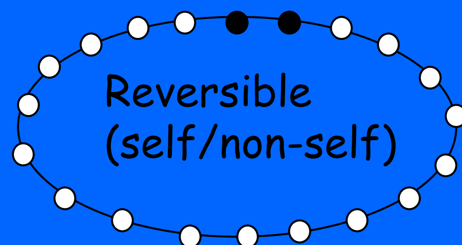
Simple Majority

Reversible (non-self)

Reversible (self)

	250	232
000	0	0
001	1	0
010	0	0
011	1	1
100	1	0
101	1	1
110	1	1
111	1	1





Strong Majority

Irreversible (self)

Reversible (non-self)

Reversible (self)

Irreversible (non-self)

	Irreversible (non-self)	Irreversible (self)	Reversible (non-self)	Reversible (self)
000	0	0	0	0
001	0	0	0	0
010	1	1	0	0
011	1	1	0	1
100	0	0	0	0
101	1	1	1	1
110	1	1	0	1
111	1	1	1	1
	236	236	160	233