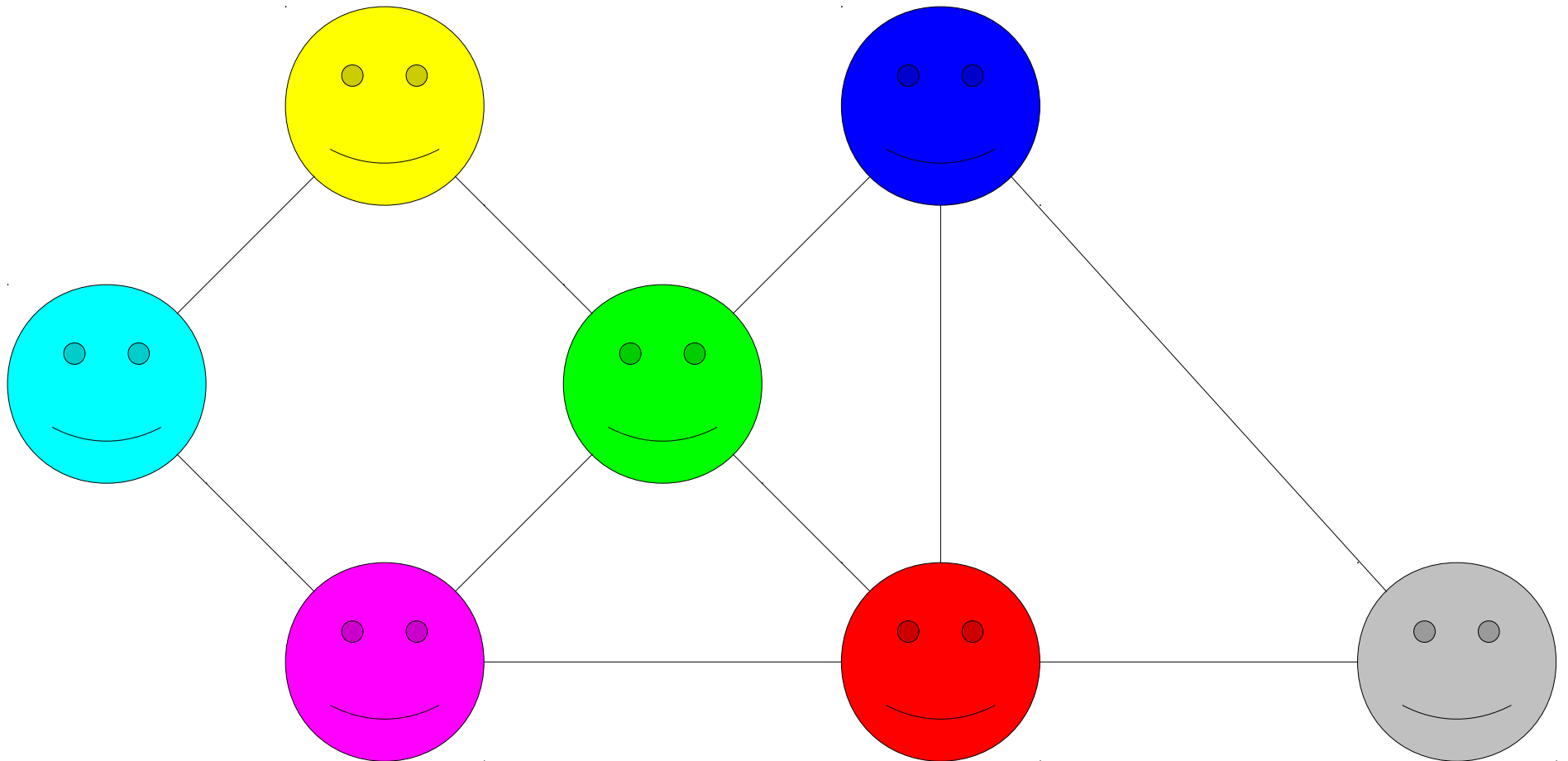
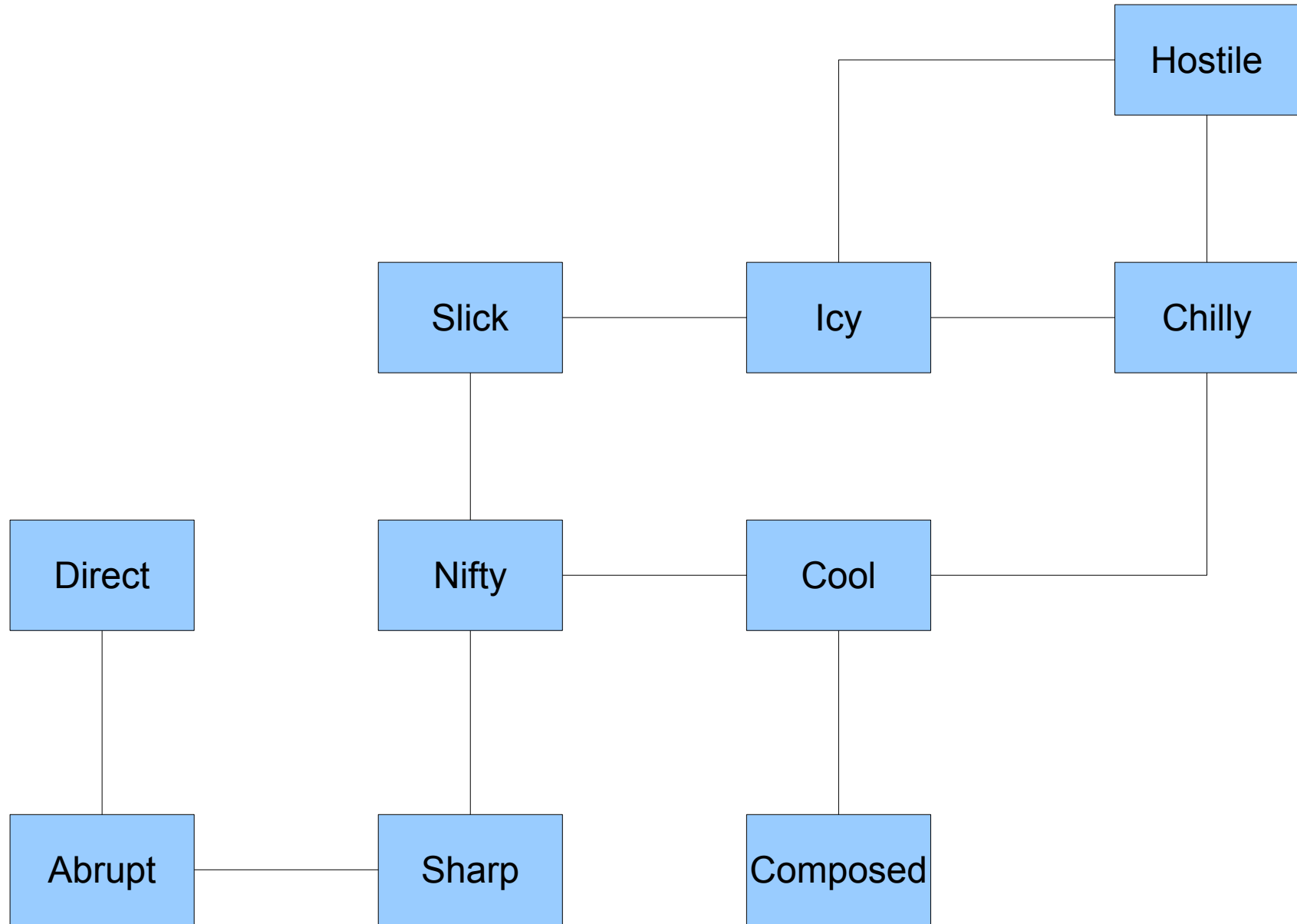


Graphs and Networks

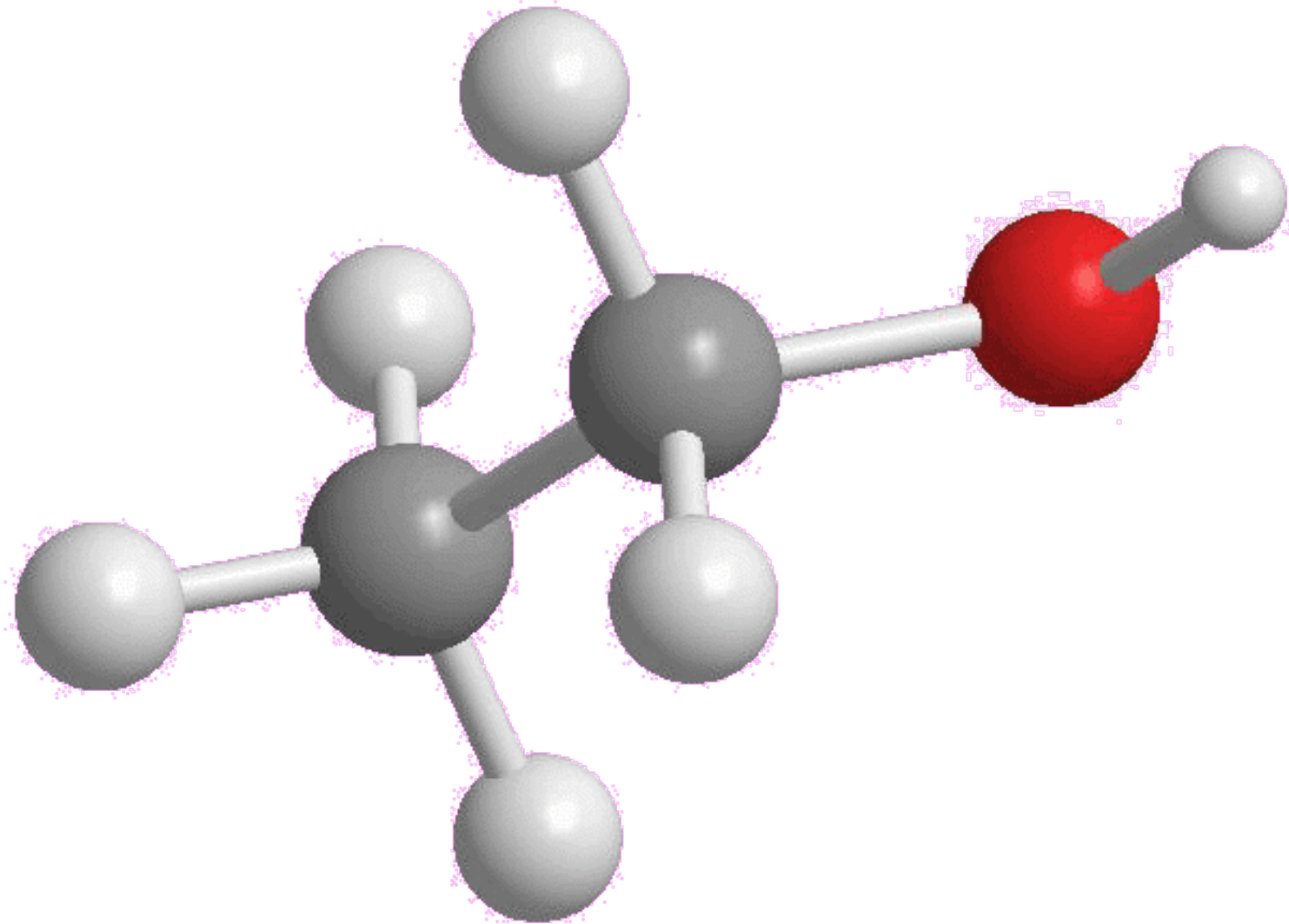
A Social Network

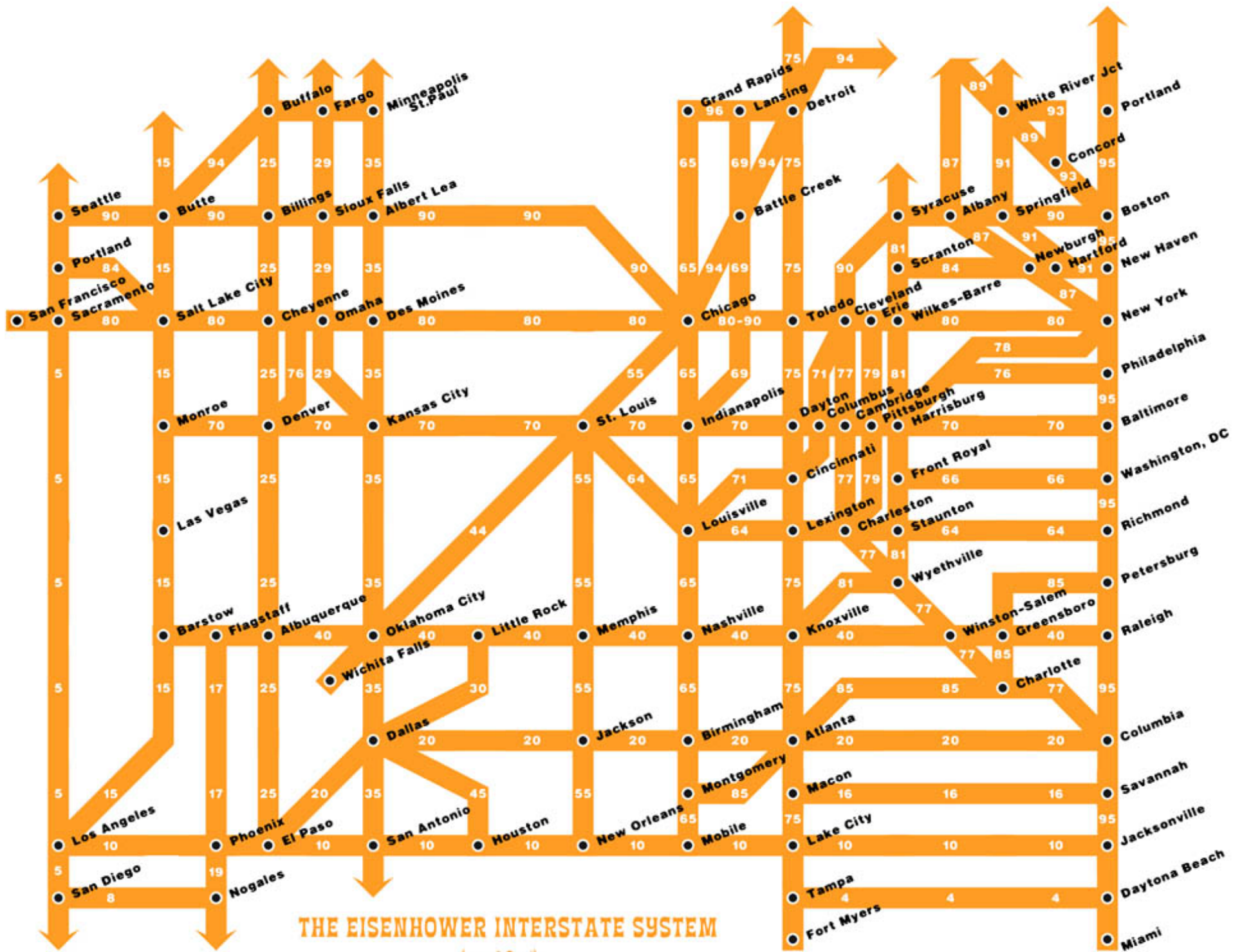


Synonyms

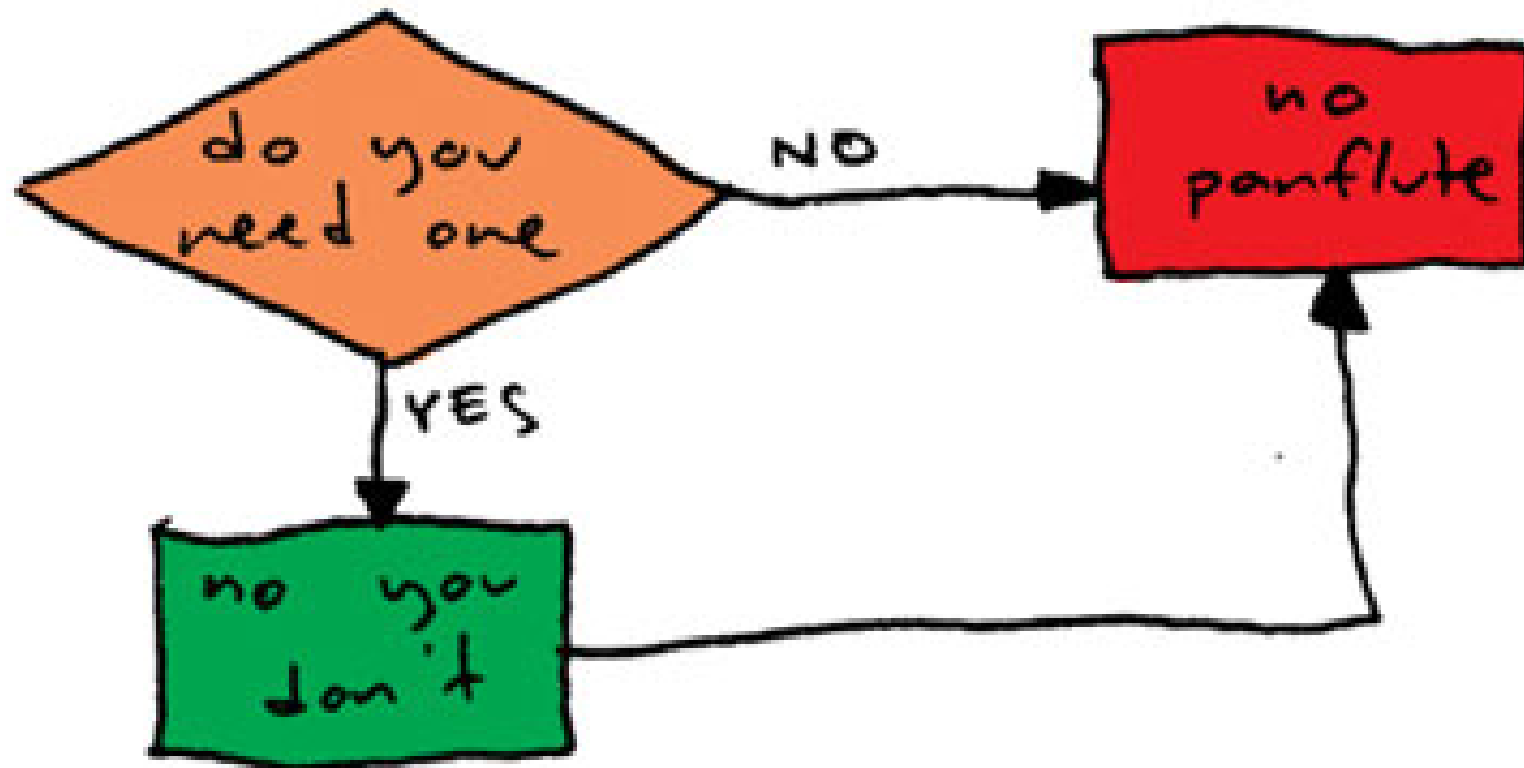


Chemical Bonds





PANFLUTE FLOWCHART

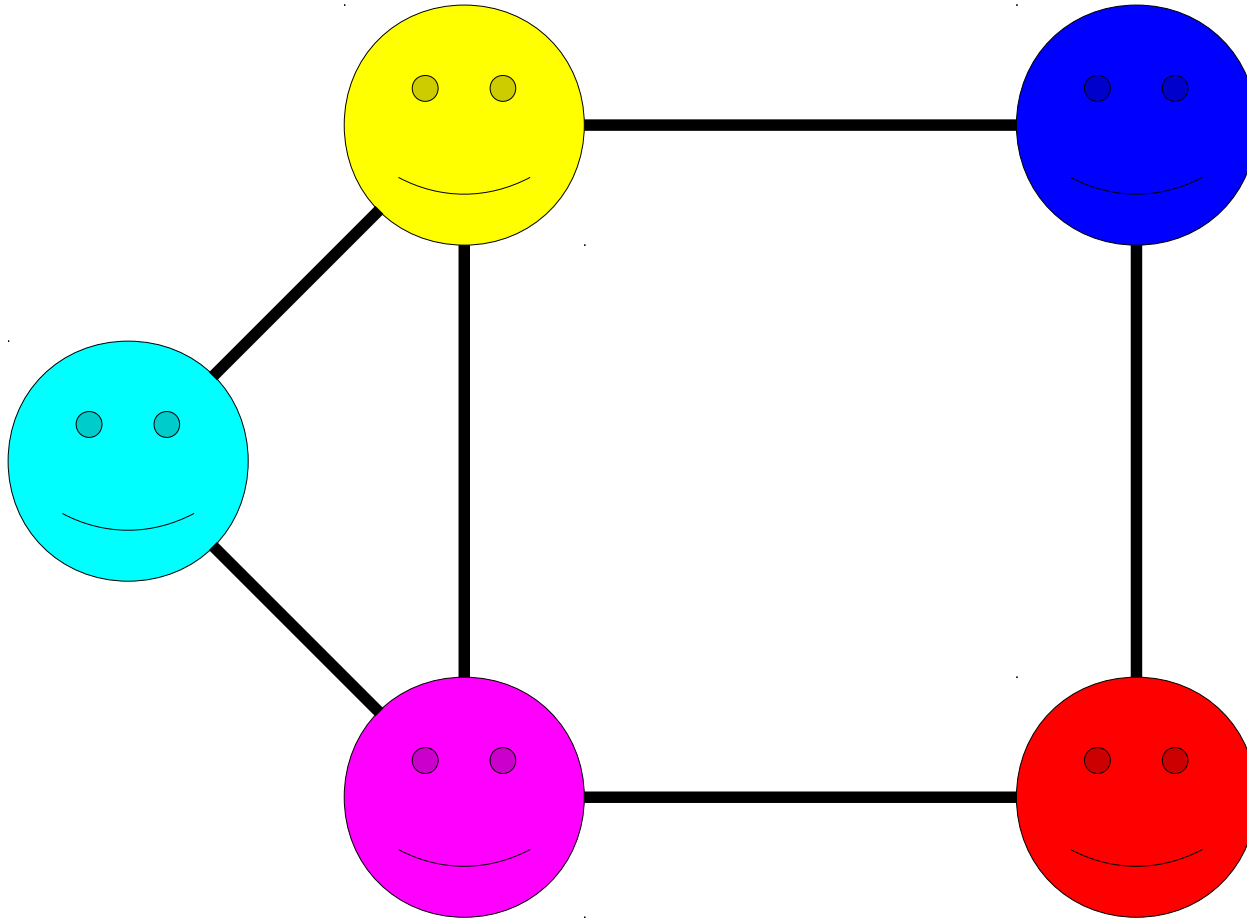




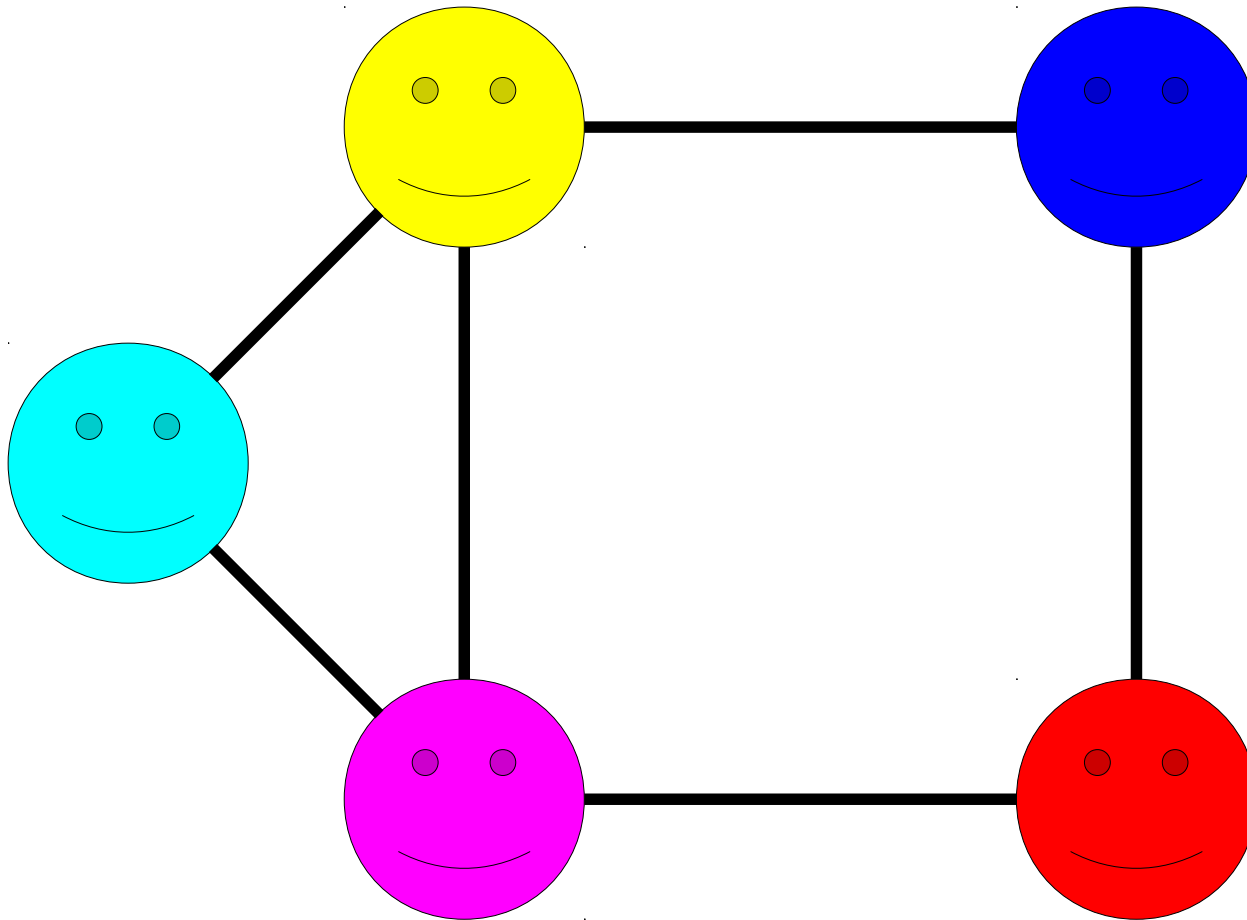
A *graph* is a mathematical structure for representing relationships.

A ***graph*** is a mathematical structure for representing relationships.

A ***graph*** is a mathematical structure for representing relationships.

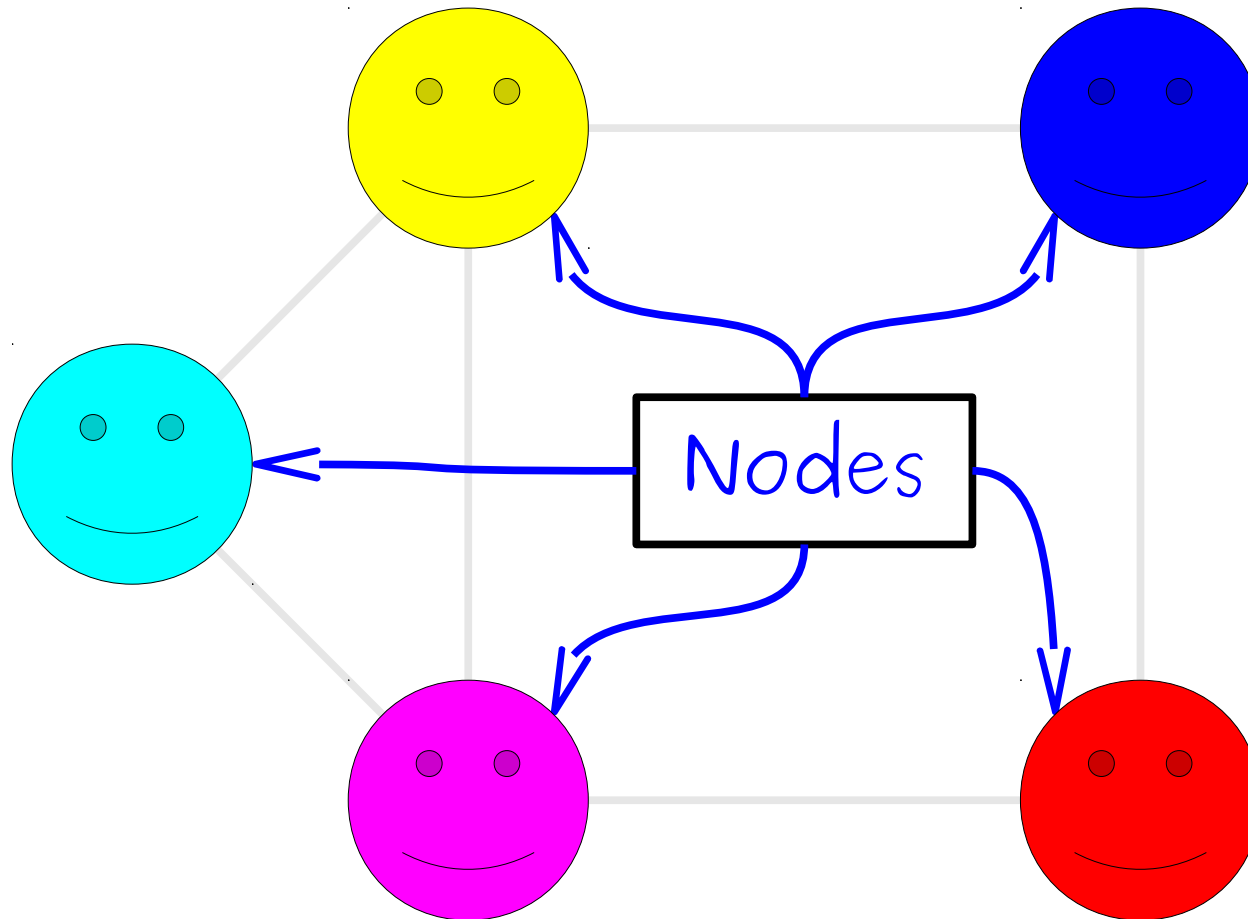


A **graph** is a mathematical structure for representing relationships.



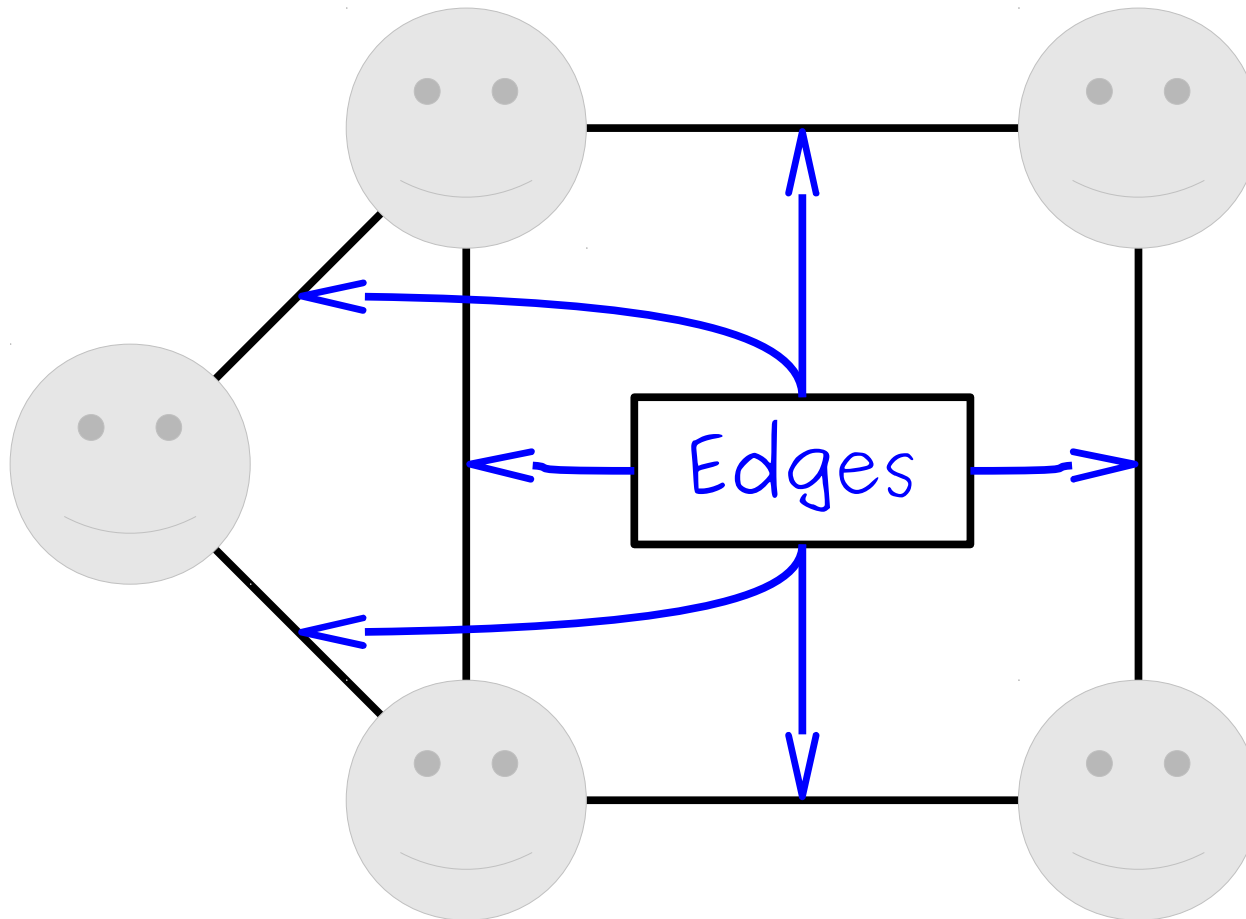
A graph consists of a set of **nodes** connected by **edges**.

A **graph** is a mathematical structure for representing relationships.



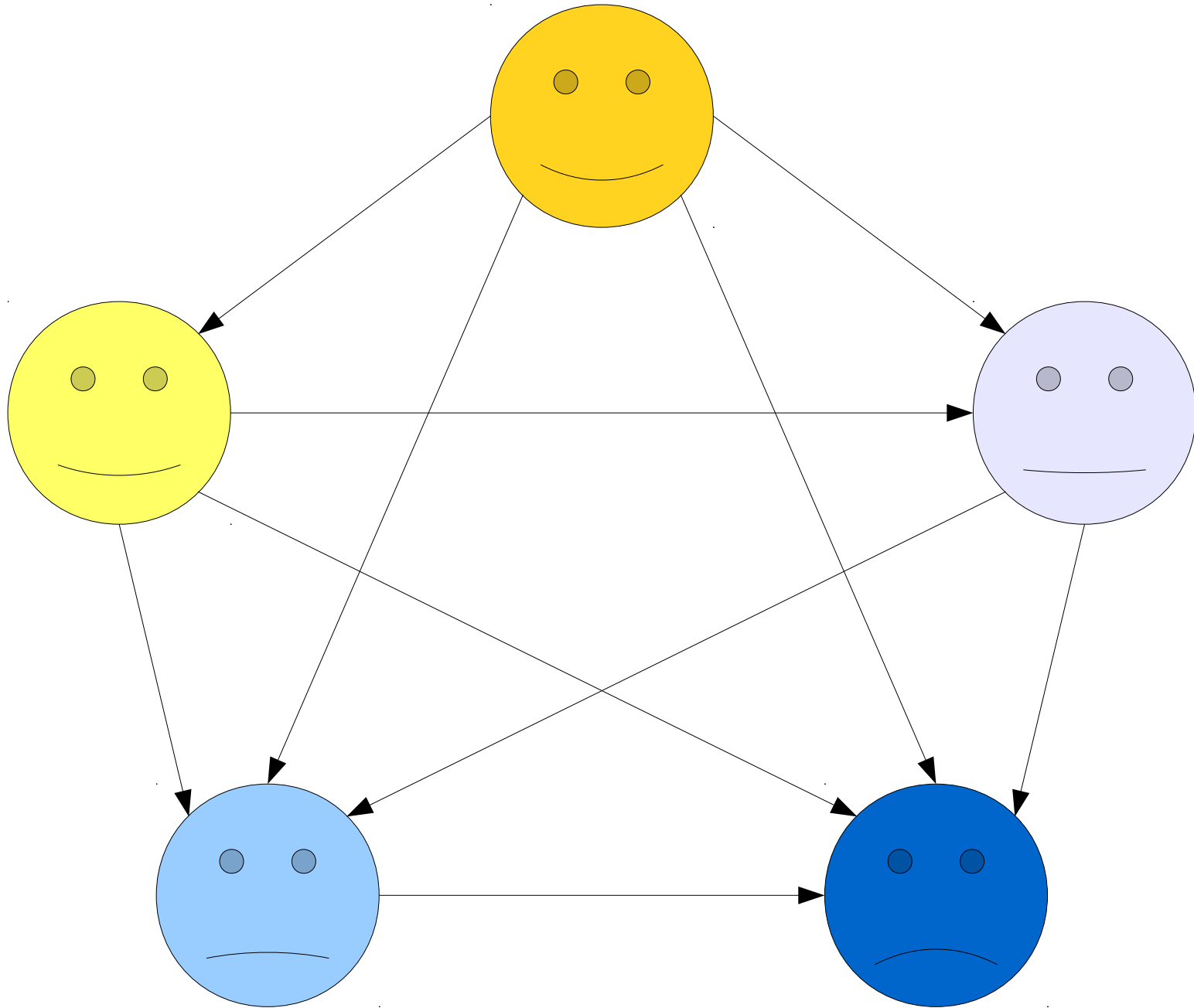
A graph consists of a set of **nodes** connected by **edges**.

A **graph** is a mathematical structure for representing relationships.

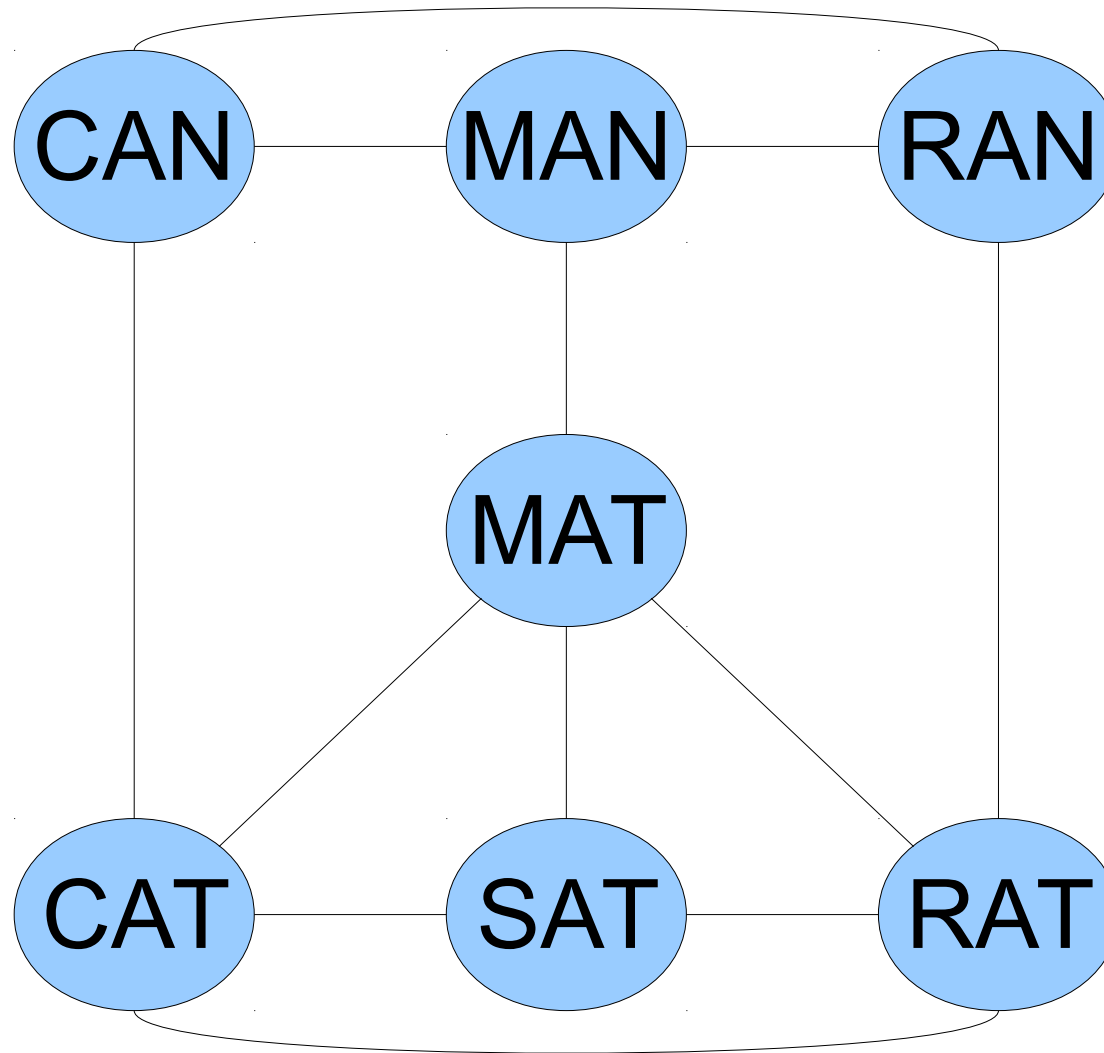


A graph consists of a set of **nodes** connected by **edges**.

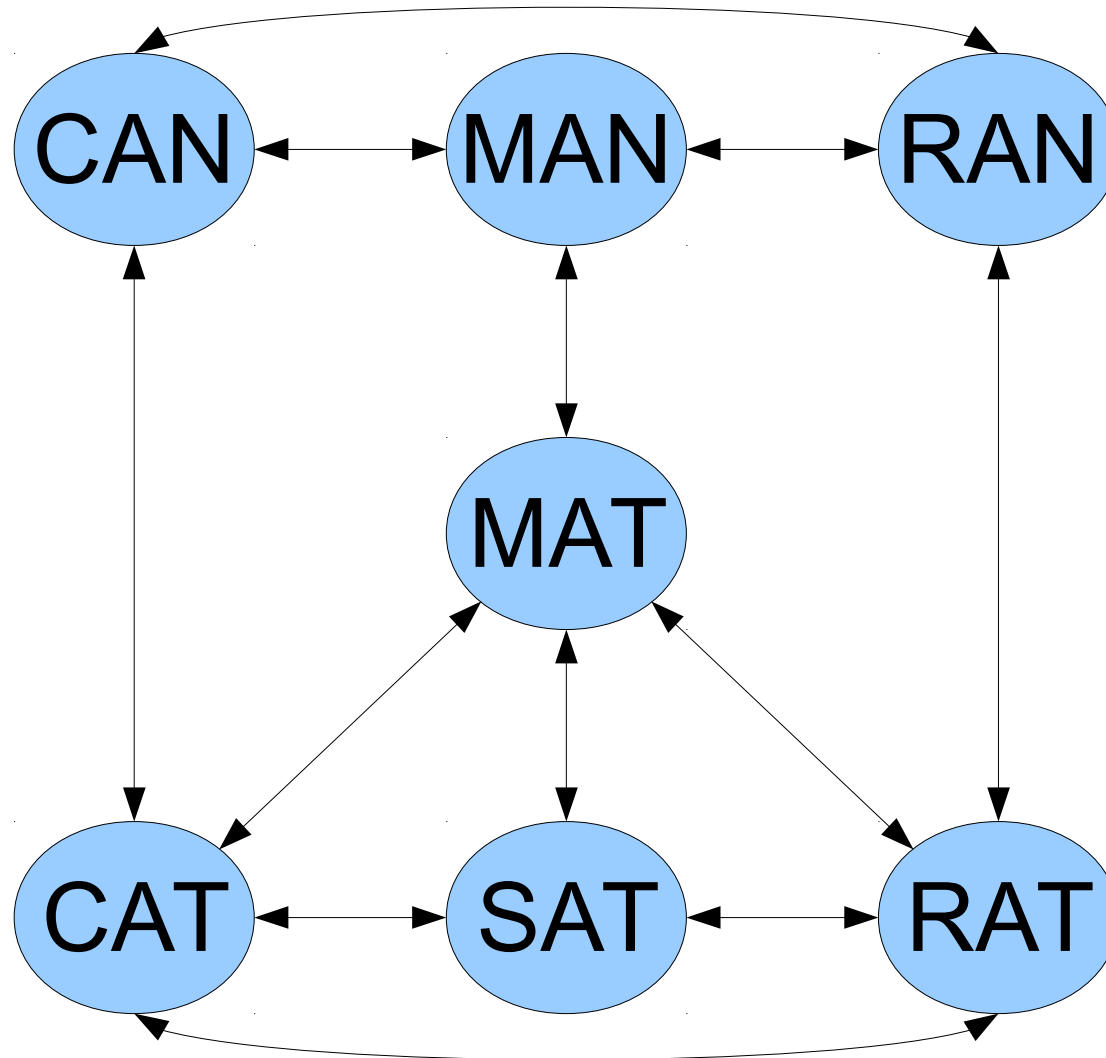
Some graphs are *directed*.



Some graphs are *undirected*.



Some graphs are *undirected*.

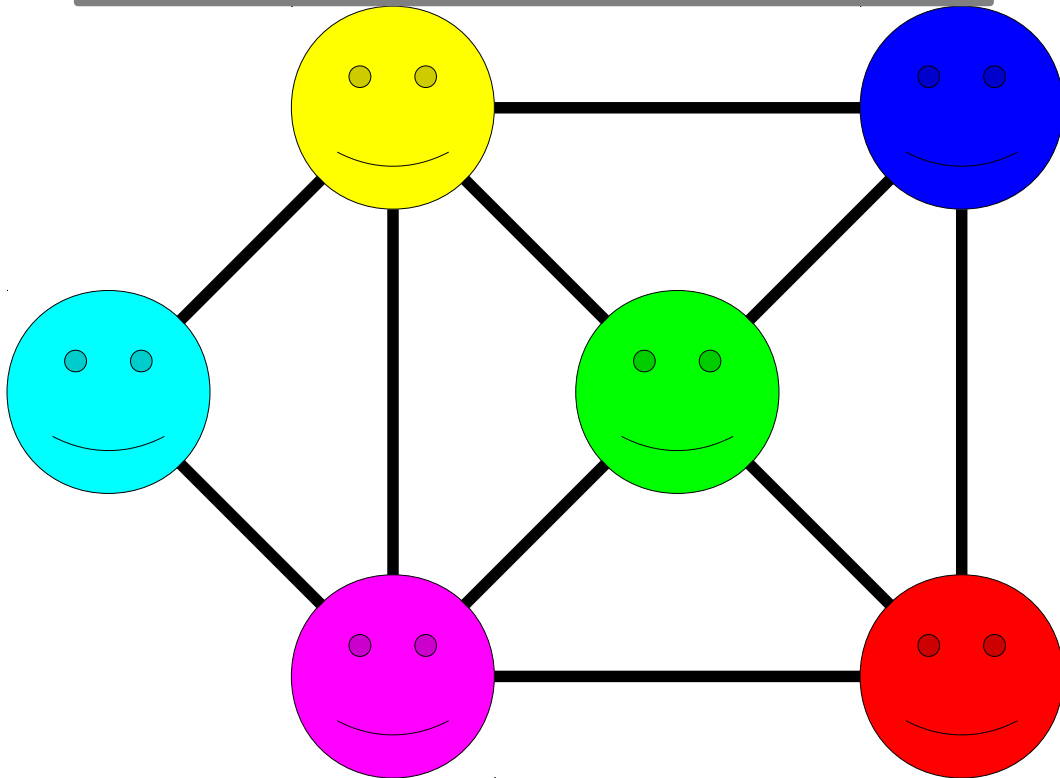


You can think of them as directed graphs with edges both ways.

How can we represent graphs in Java?

Representing Graphs

We can represent a graph as a map from nodes to the list of nodes each node is connected to.



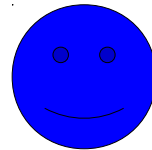
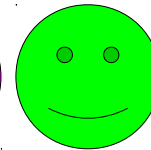
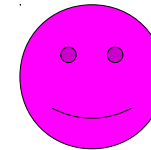
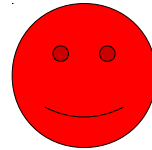
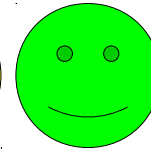
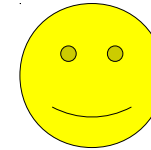
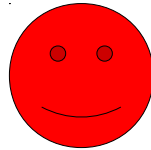
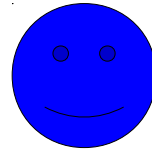
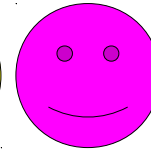
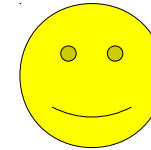
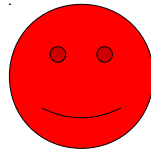
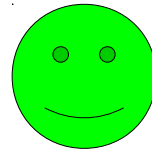
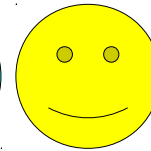
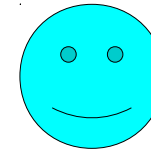
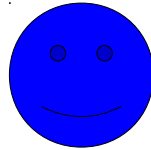
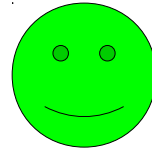
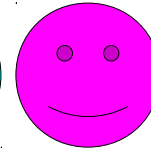
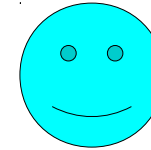
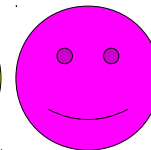
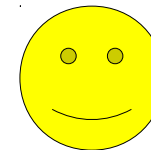
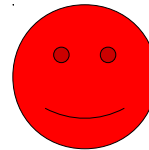
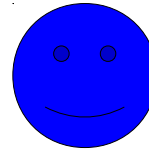
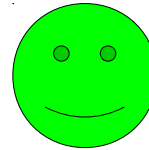
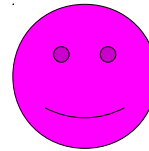
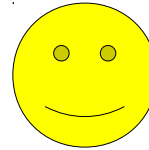
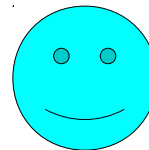
HashMap<**Node**, ArrayList<**Node**>>

Node

ArrayList<**Node**>

Node

Connected To



The Wikipedia Graph



WIKIPEDIA
The Free Encyclopedia

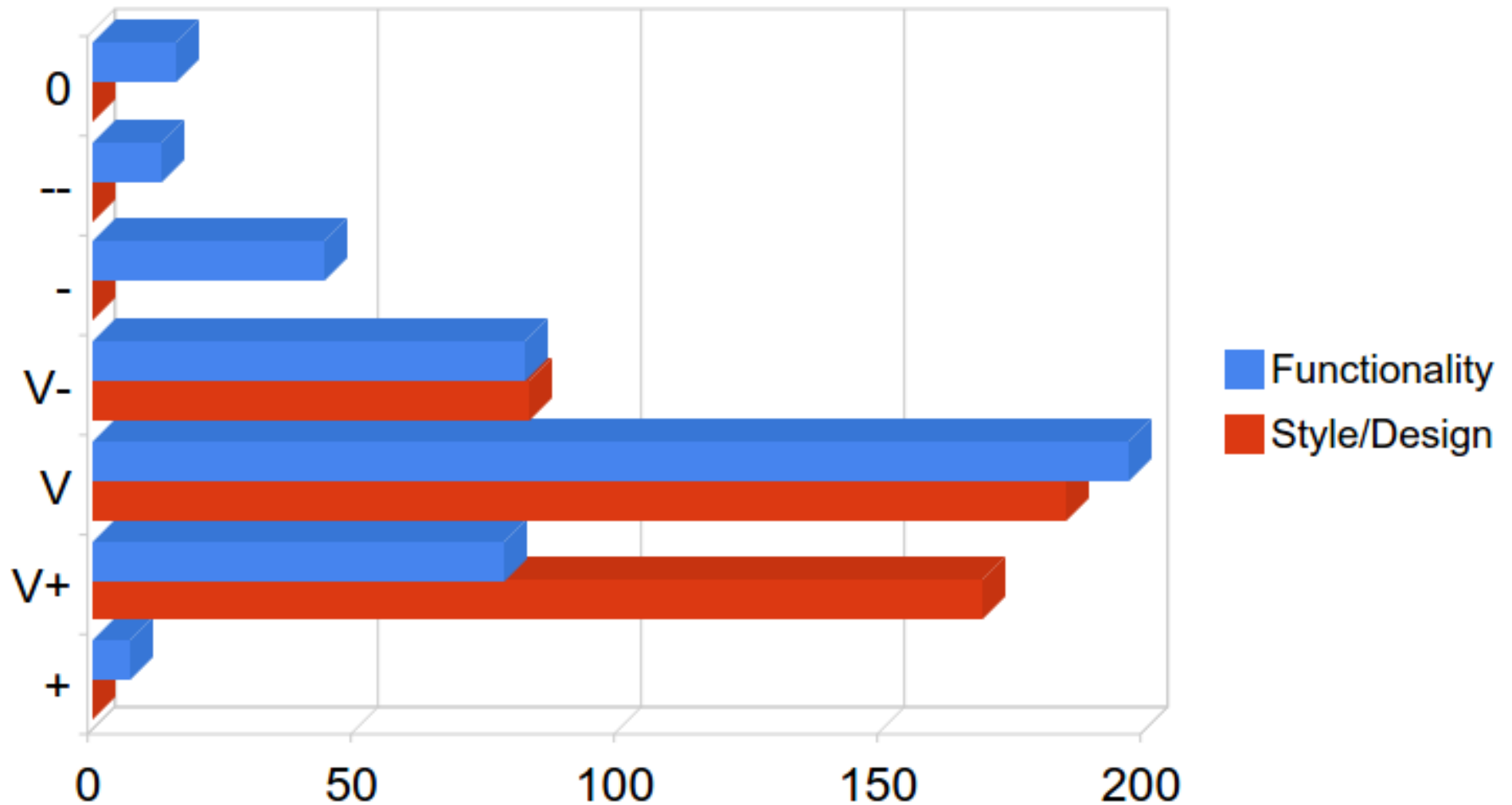
- Wikipedia (and the web in general) is a graph!
- Each page is a node.
- There is an edge from one page to another if the first page links to the second.

Time Out for Announcements!

General Announcements

- Assignment 6 due Friday at 3:15PM.
- Have questions?
 - Stop by the LaIR!
 - Email your section leader!
 - Stop by Keith's or Alisha's office hours!
- Exam regrades due at 4:15PM today - hand them to Keith or Alisha.

Assignment 5 Grades



Midterm Logistics

- Second midterm is next **Tuesday, March 3** from **7PM - 10PM**, locations TBA.
- Closed-book, closed-computer, open-one-page-of-notes.
 - Reference sheet will be provided; available up on the course website.
- Topic coverage:
 - Covers material up through and including today's lecture on graphs.
 - Will *not* cover interactors or Karel the Robot.
 - Focus is on material from Assignment 5 and Assignment 6, though by nature will be cumulative.
- Need to take the exam at an alternate time? Contact Alisha by tonight.

Preparing for the Exam

- Practice exam **tomorrow (Thursday) night** from **7PM - 10PM** in **Cubberly Auditorium**.
- Our recommendation:
 - Take the practice exam under realistic conditions. If you can't make it on Thursday night, find a group of fellow CS106Aers and hold your own practice exam. Trust me - this is hugely valuable!
 - Type all of your solutions into a computer and debug them until they work correctly. What sorts of corrections did you have to make?
 - Look over the reference solutions. Do you understand how they work? Type them up and try modifying them - do you see why all the pieces are there?
 - Rewrite your old assignments and problems from section handouts on paper, then repeat the above process.
 - Read over the textbook to see new patterns and learn more about concepts.

Back to CS106A!

Network Analysis

- We can analyze how nodes in a graph are connected to learn more about the graph.
- Sample questions:
 - How connected are the nodes in the graph?
 - How “important” is each node in the graph?
 - How “resilient” is the graph?

Network Analysis

We can analyze how nodes in a graph are connected to learn more about the graph.

Sample questions:

How connected are the nodes in the graph?

- How “important” is each node in the graph?

How “resilient” is the graph?

Finding Important Nodes

- Suppose that we want to have the computer find “important” articles on Wikipedia.
- We just have the link structure, not the text of the page, the number of edits, the length of the article, etc.
- How might we do this?

Link Analysis

- To find important Wikipedia pages, let's look at the links between pages.
- We'll make two assumptions:
 - The more important an article is, the more pages will link to it.
 - The more important an article is, the more that its links matter.
- ***An article is important if other important articles link to it.***

Link Analysis

To find important Wikipedia pages, let's look at the links between pages.

We'll make two assumptions:

The more important an article is, the more pages will link to it.

The more important an article is, the more that its links matter.

- ***An article is important if other important articles link to it.***

Link Analysis



- ***An article is important if other important articles link to it.***

Link Analysis

To find
look at
We'll r
The r
page
The r
that



let's
more
more

- ***An article is important if other important articles link to it.***

The Random Surfer Model

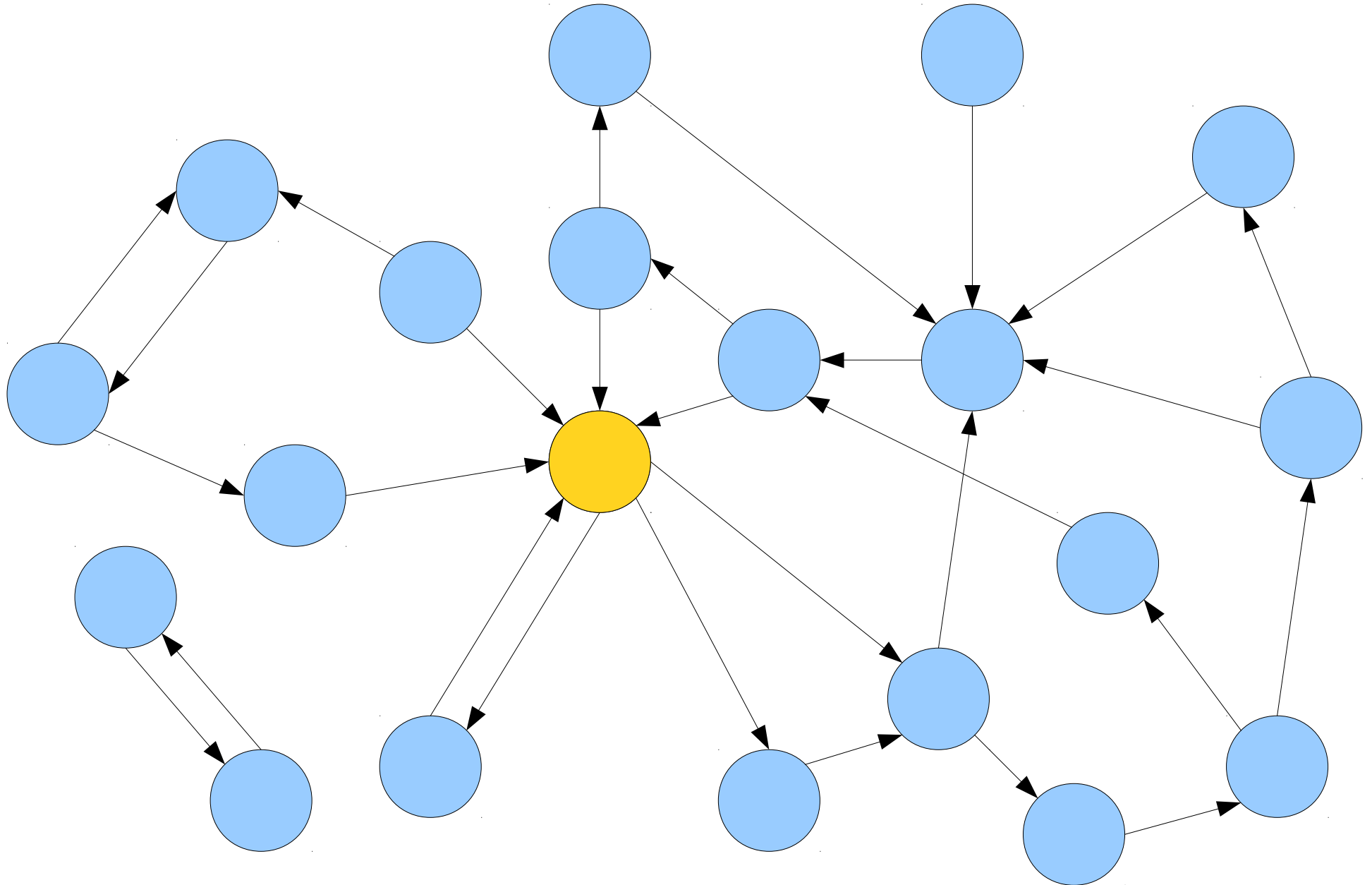


(seriously though)

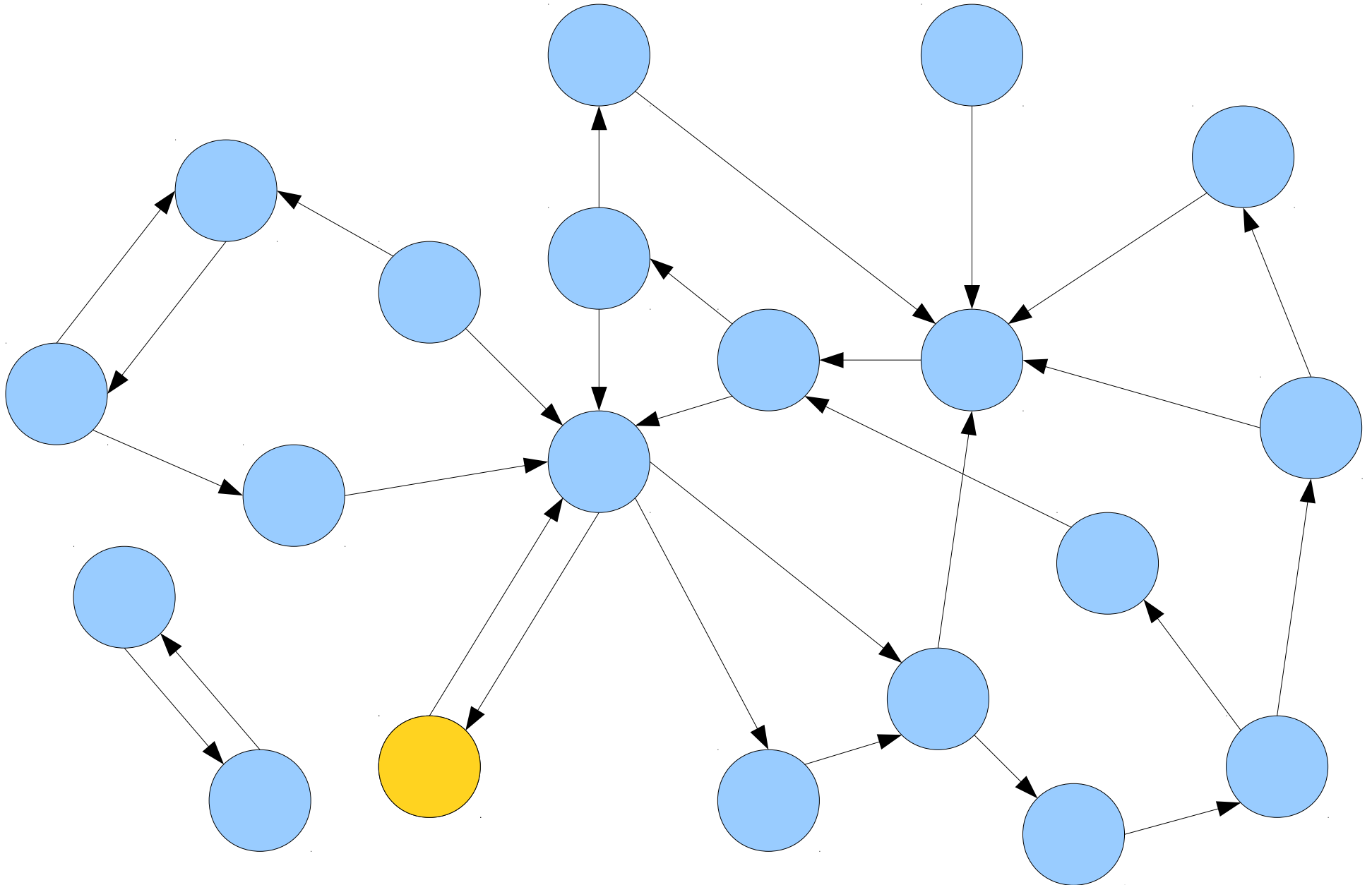
The Random Surfer Model

- Think about the behavior of a Wikipedia reader who randomly surfs Wikipedia.
- Visits some initial page at random.
- From there, the user either
 - Clicks a random link on the page to some other article, or
 - hits the “random page” link to visit a totally random page.

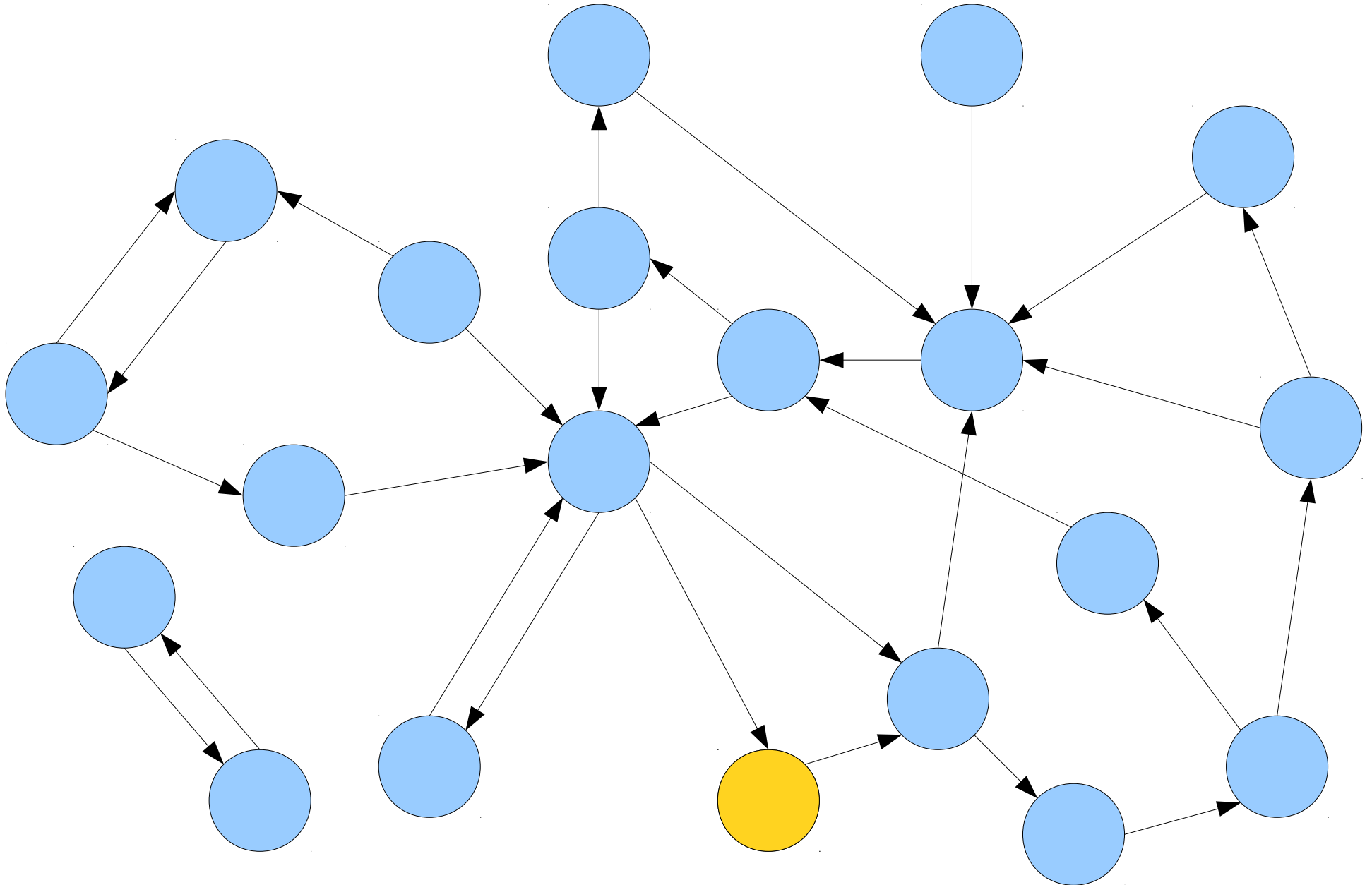
The Random Surfer Model



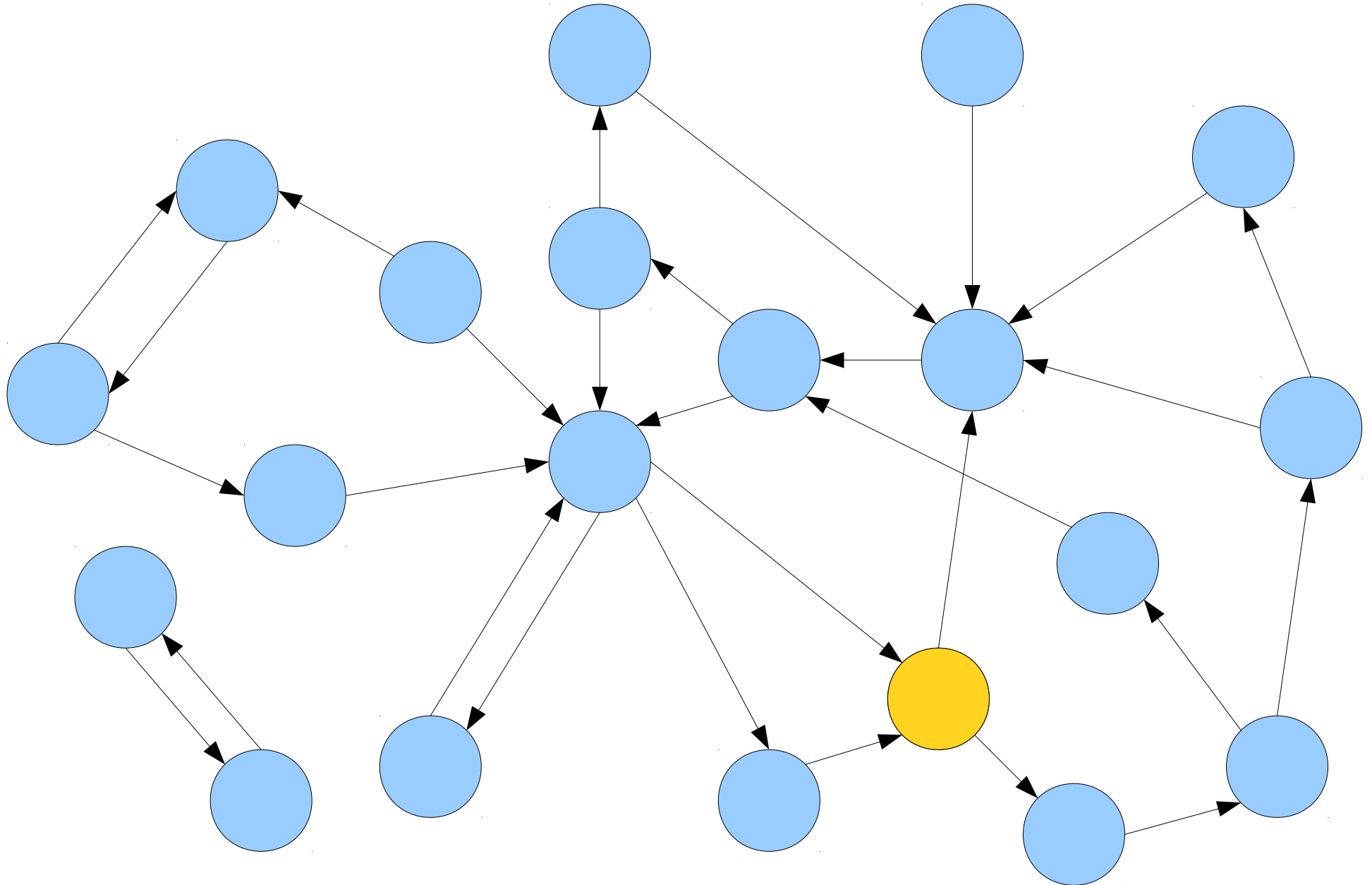
The Random Surfer Model



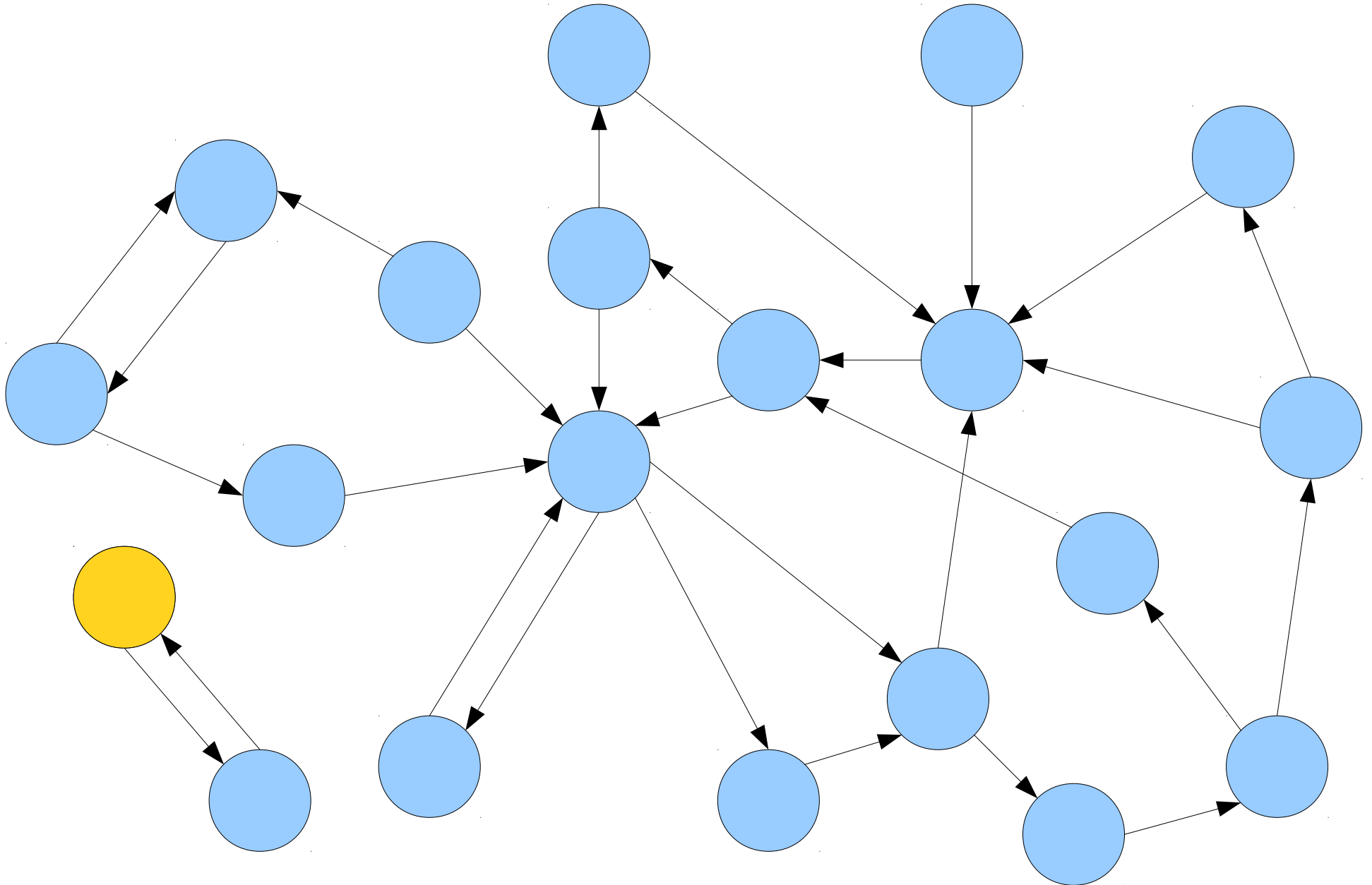
The Random Surfer Model



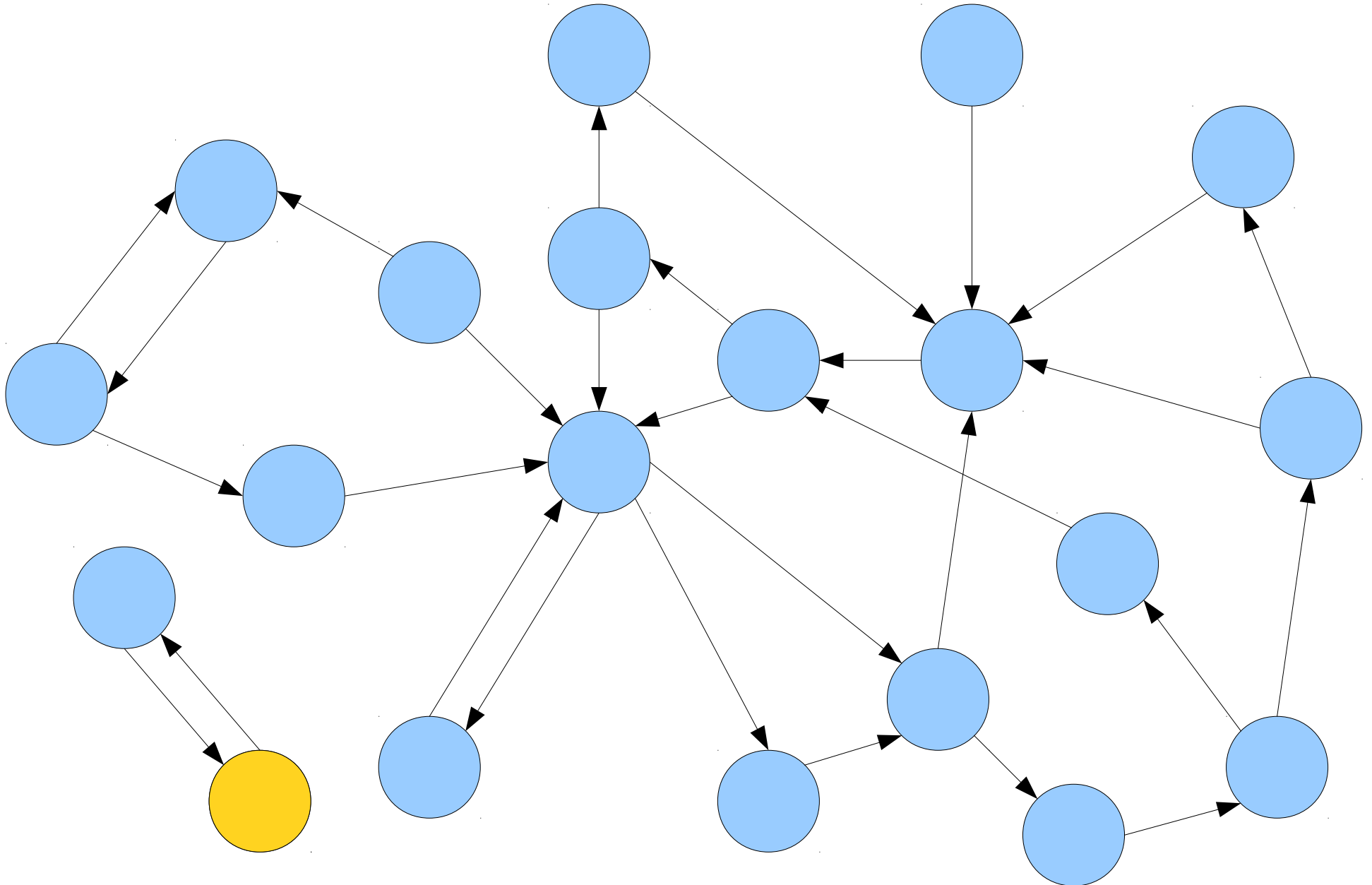
The Random Surfer Model



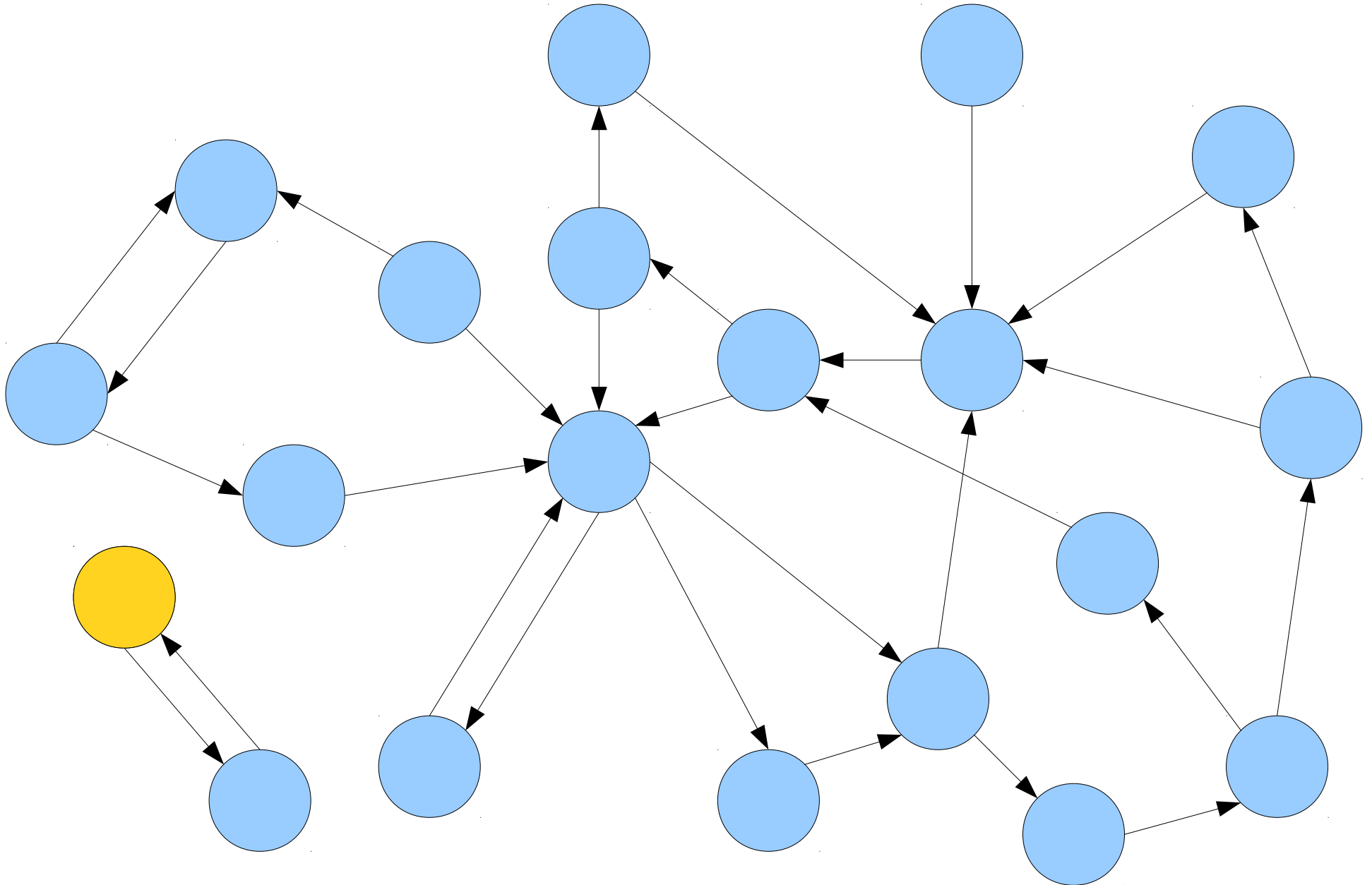
The Random Surfer Model



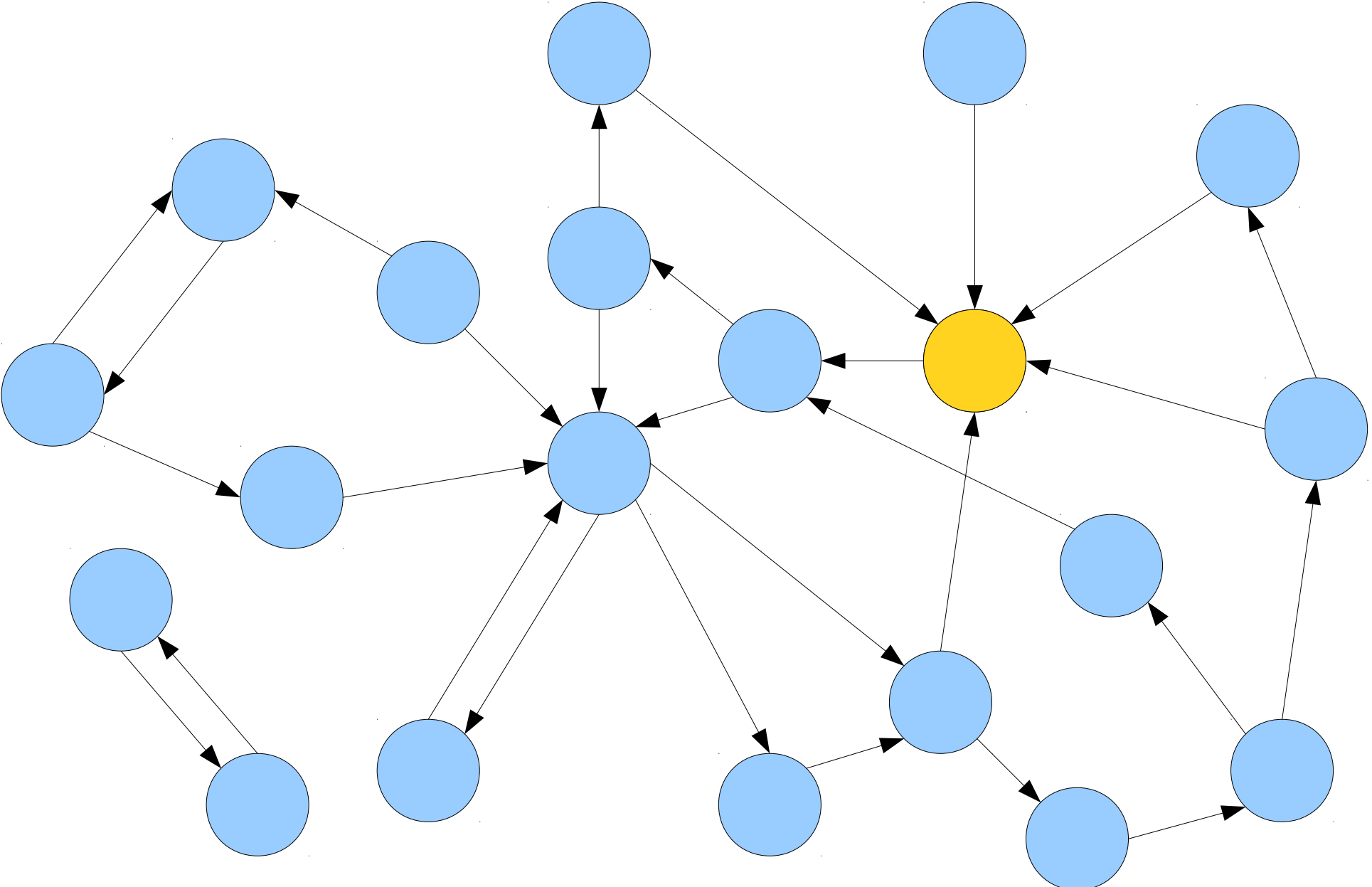
The Random Surfer Model



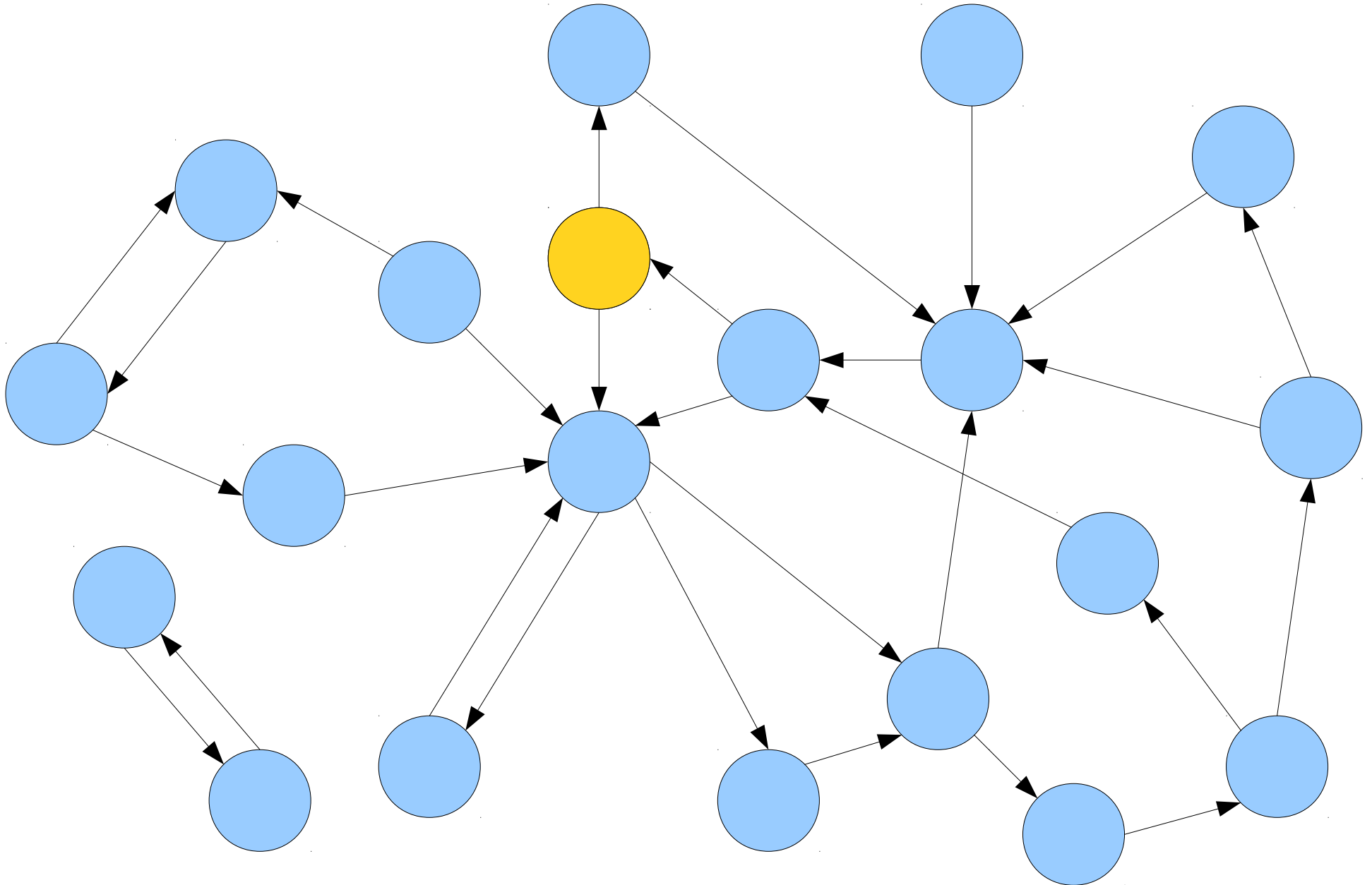
The Random Surfer Model



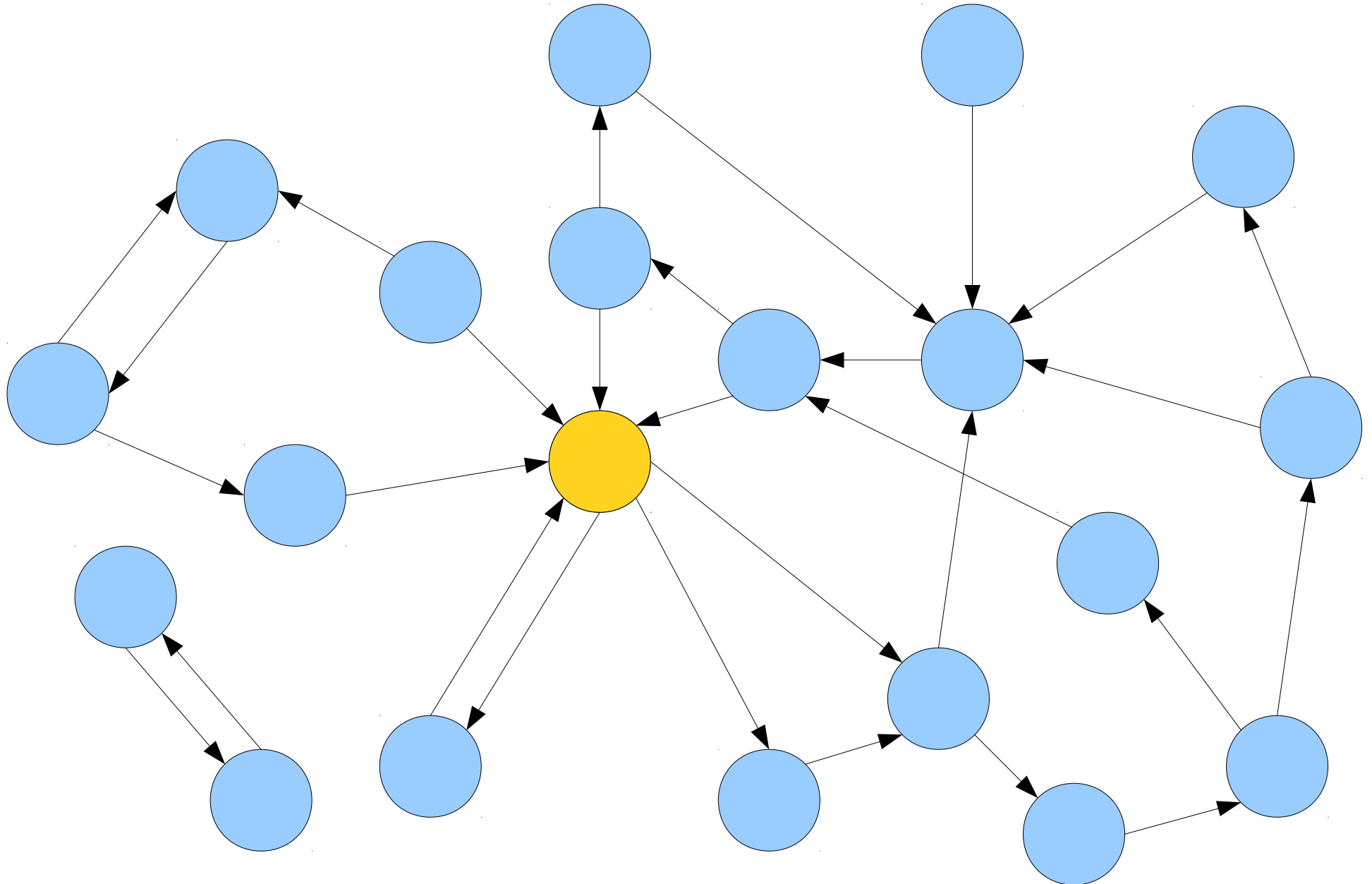
The Random Surfer Model



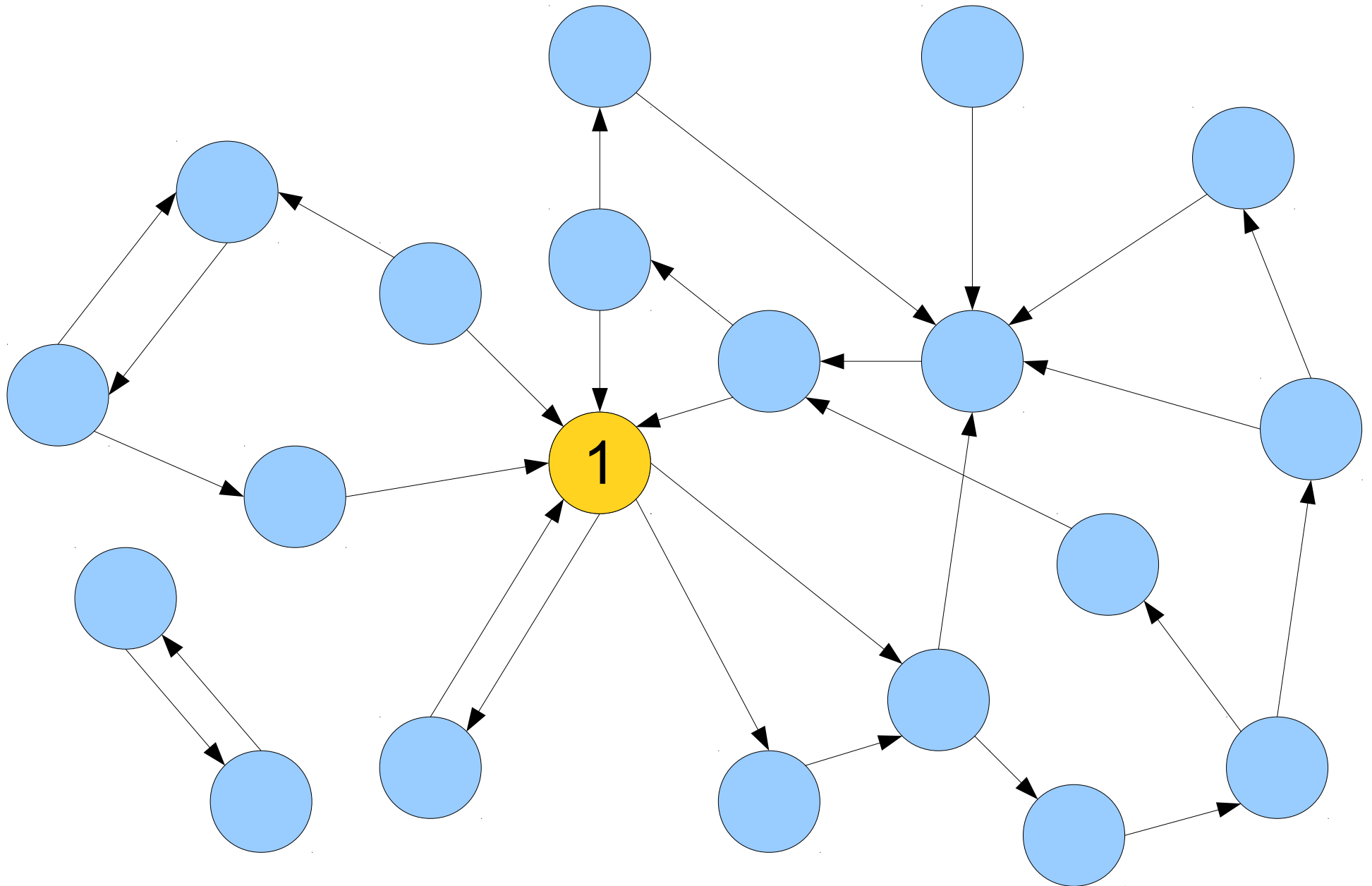
The Random Surfer Model



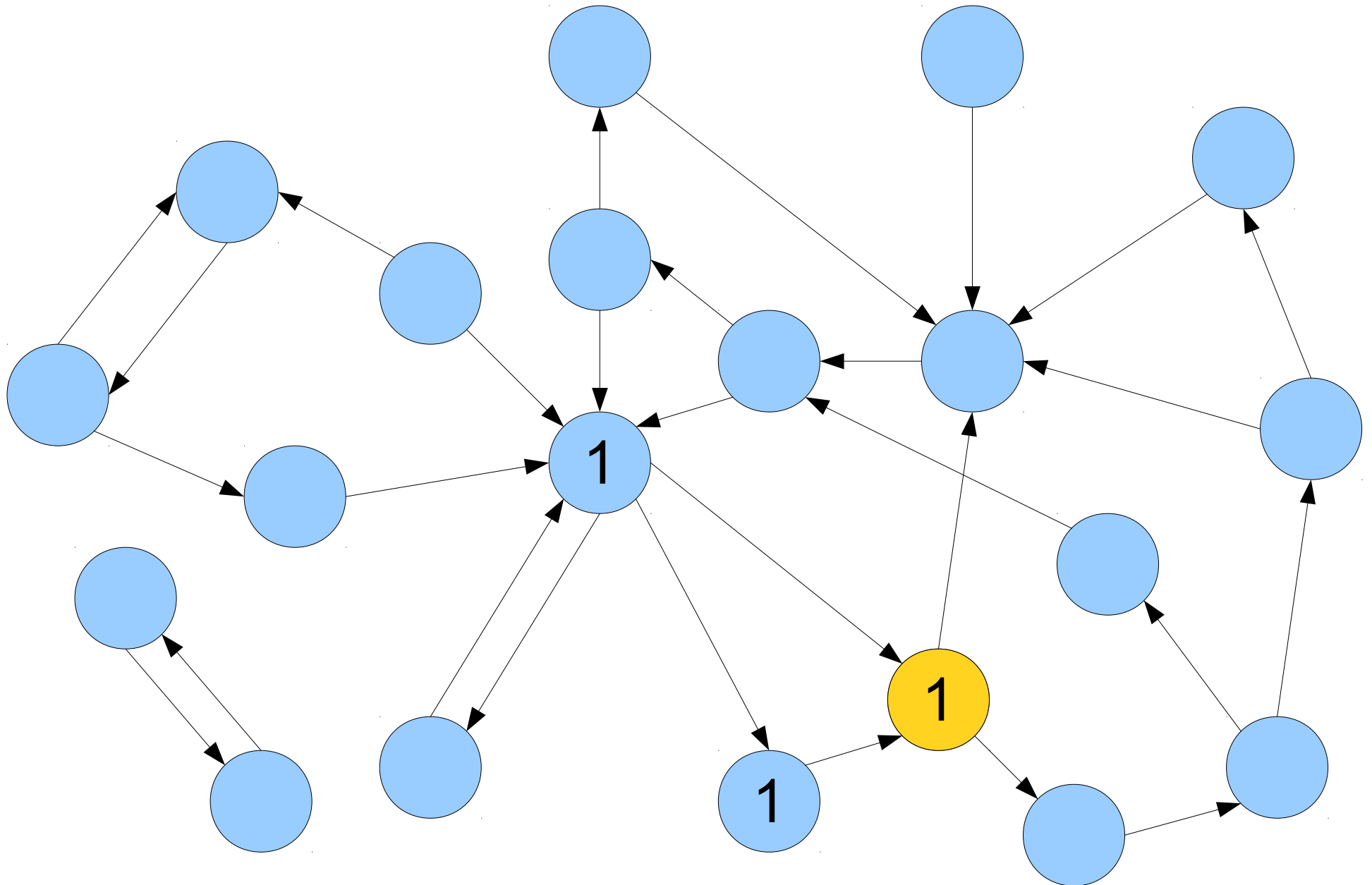
The Random Surfer Model



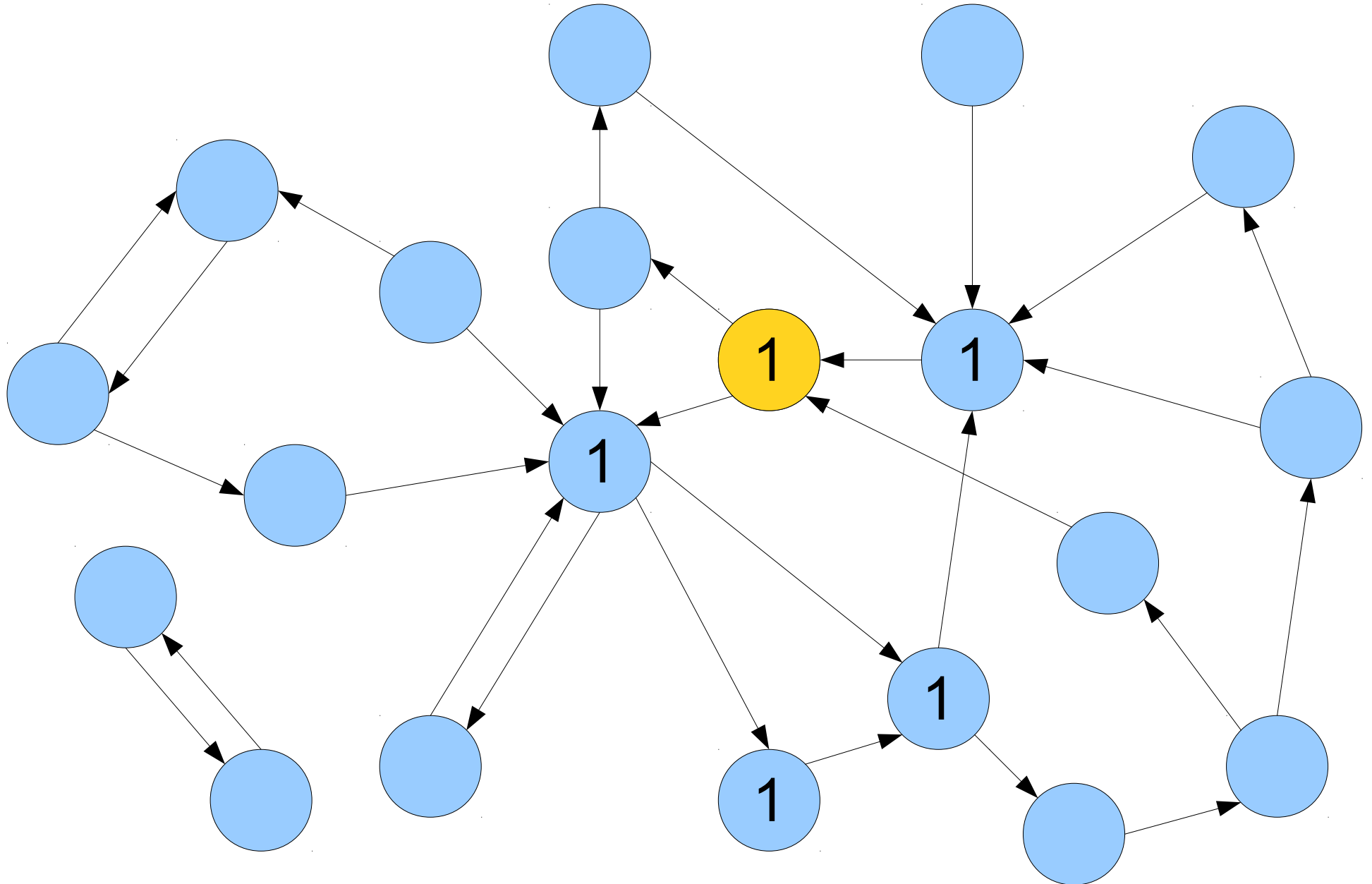
The Random Surfer Model



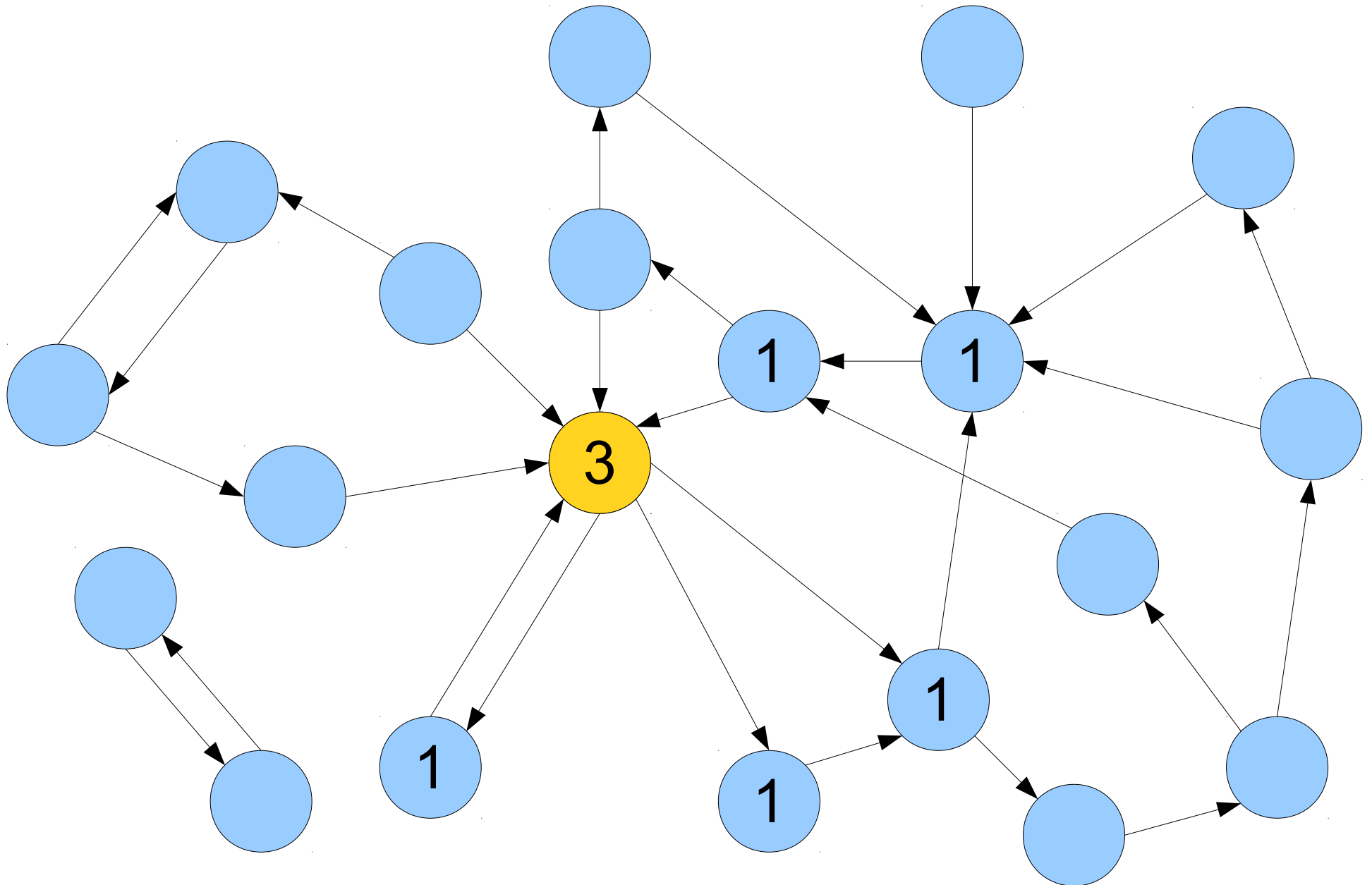
The Random Surfer Model



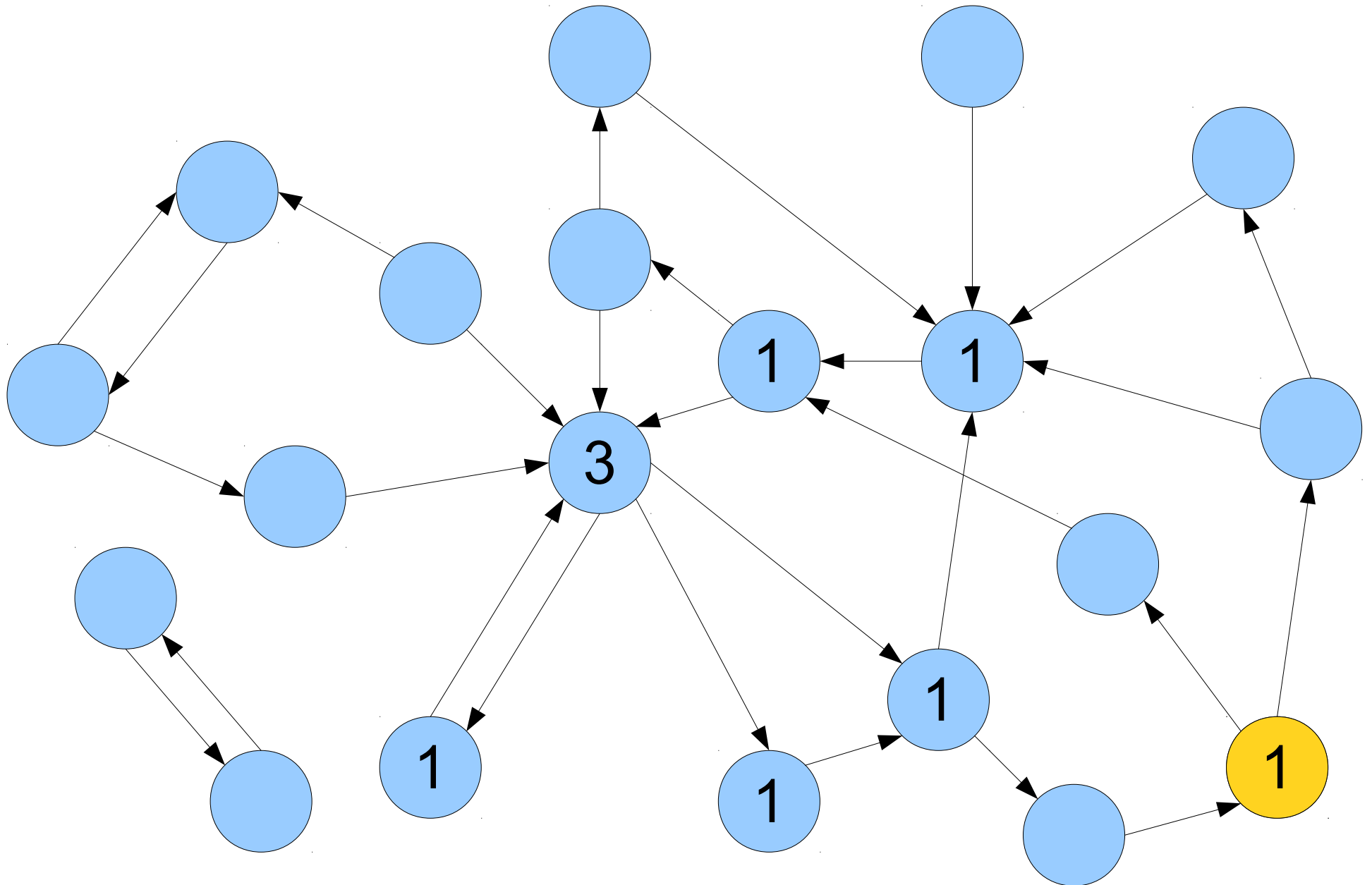
The Random Surfer Model



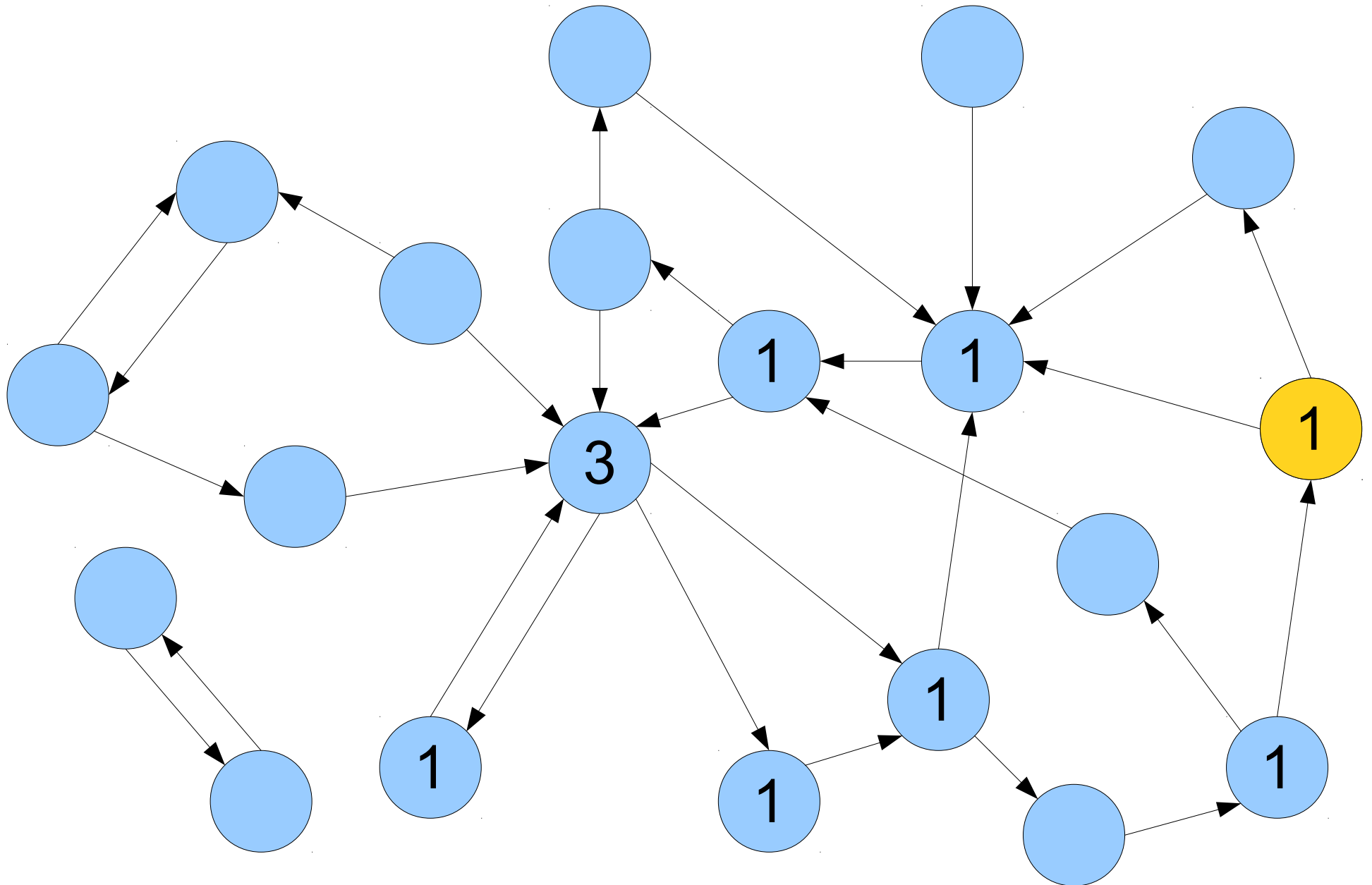
The Random Surfer Model



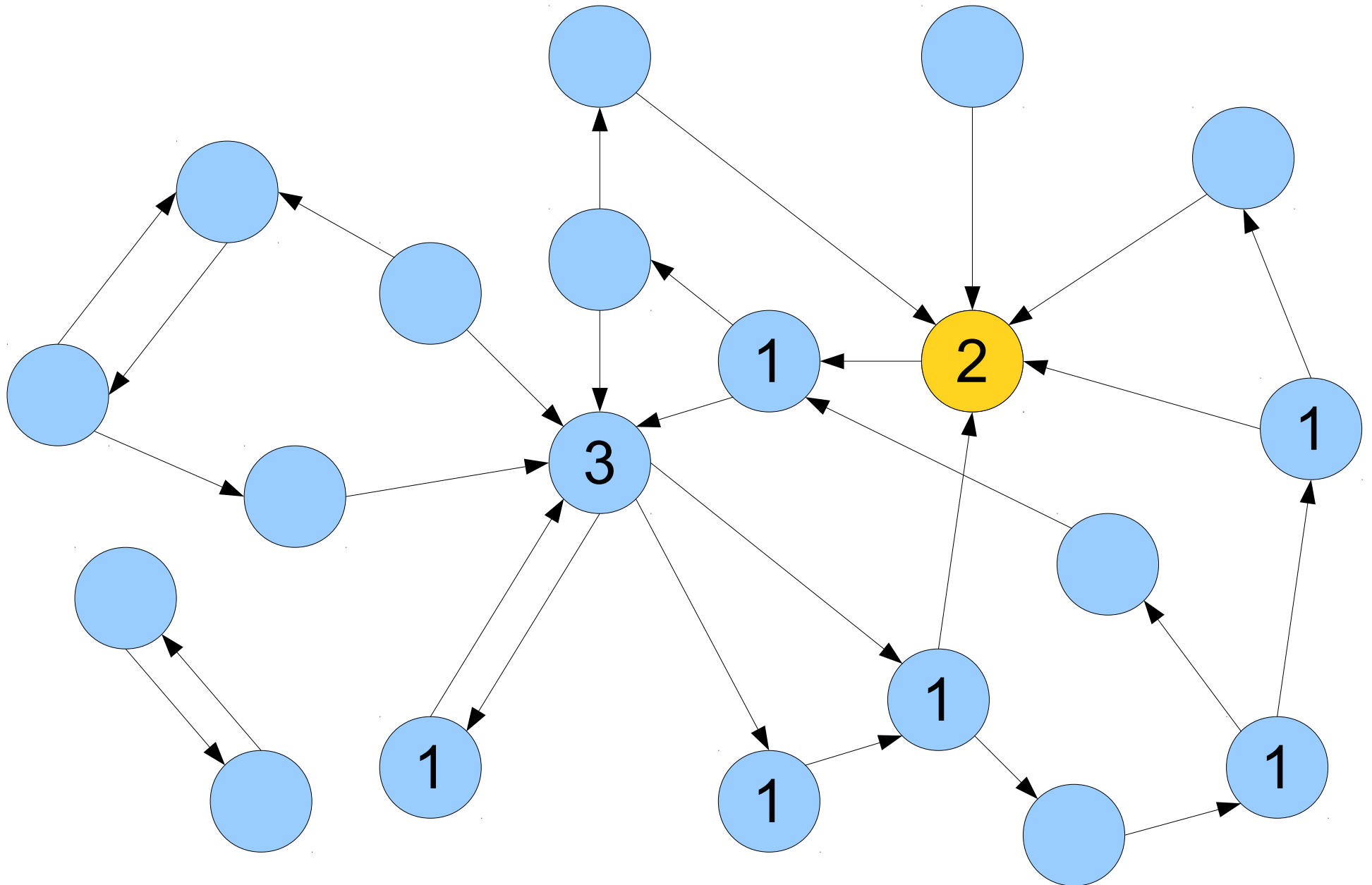
The Random Surfer Model



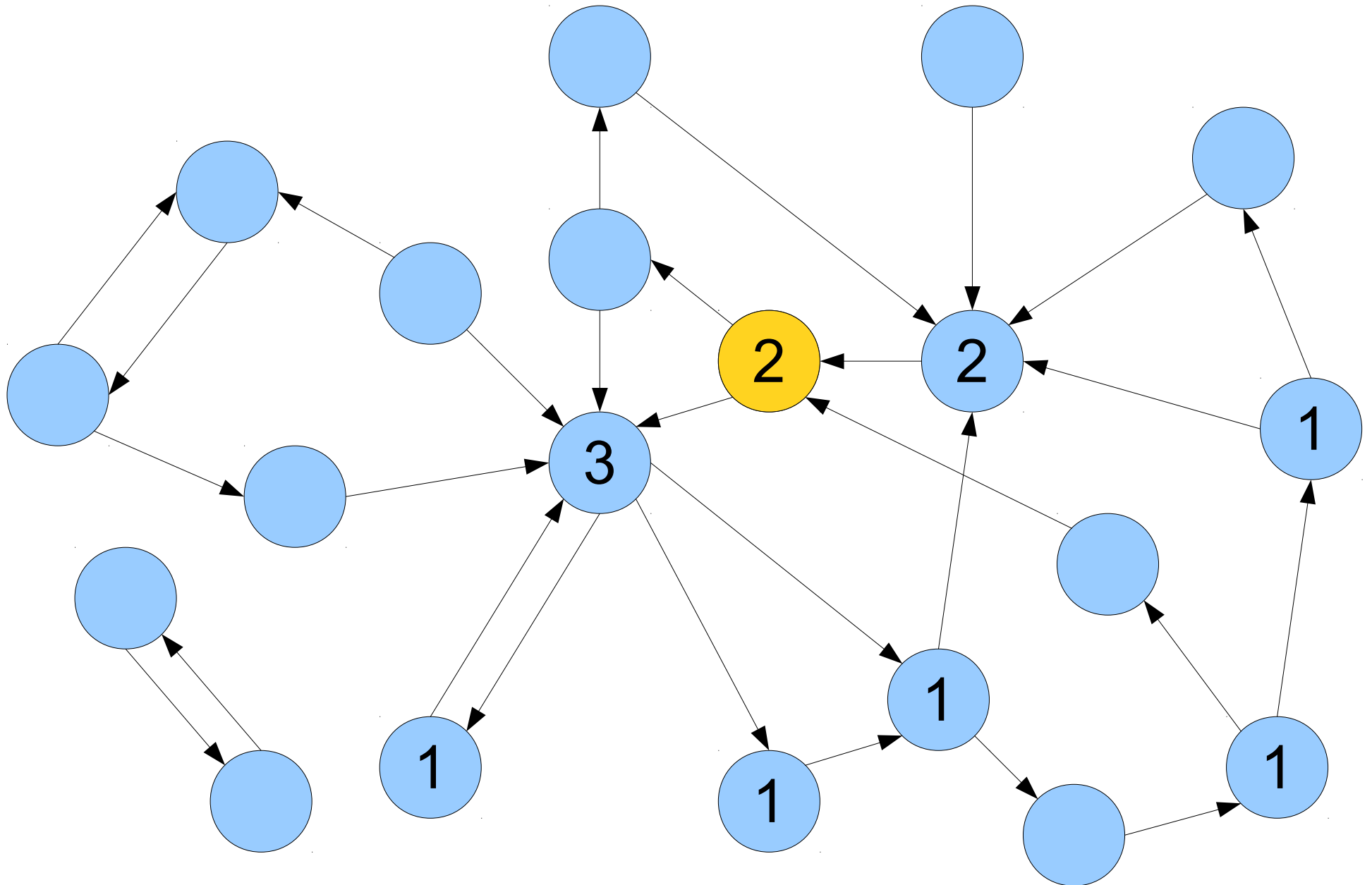
The Random Surfer Model



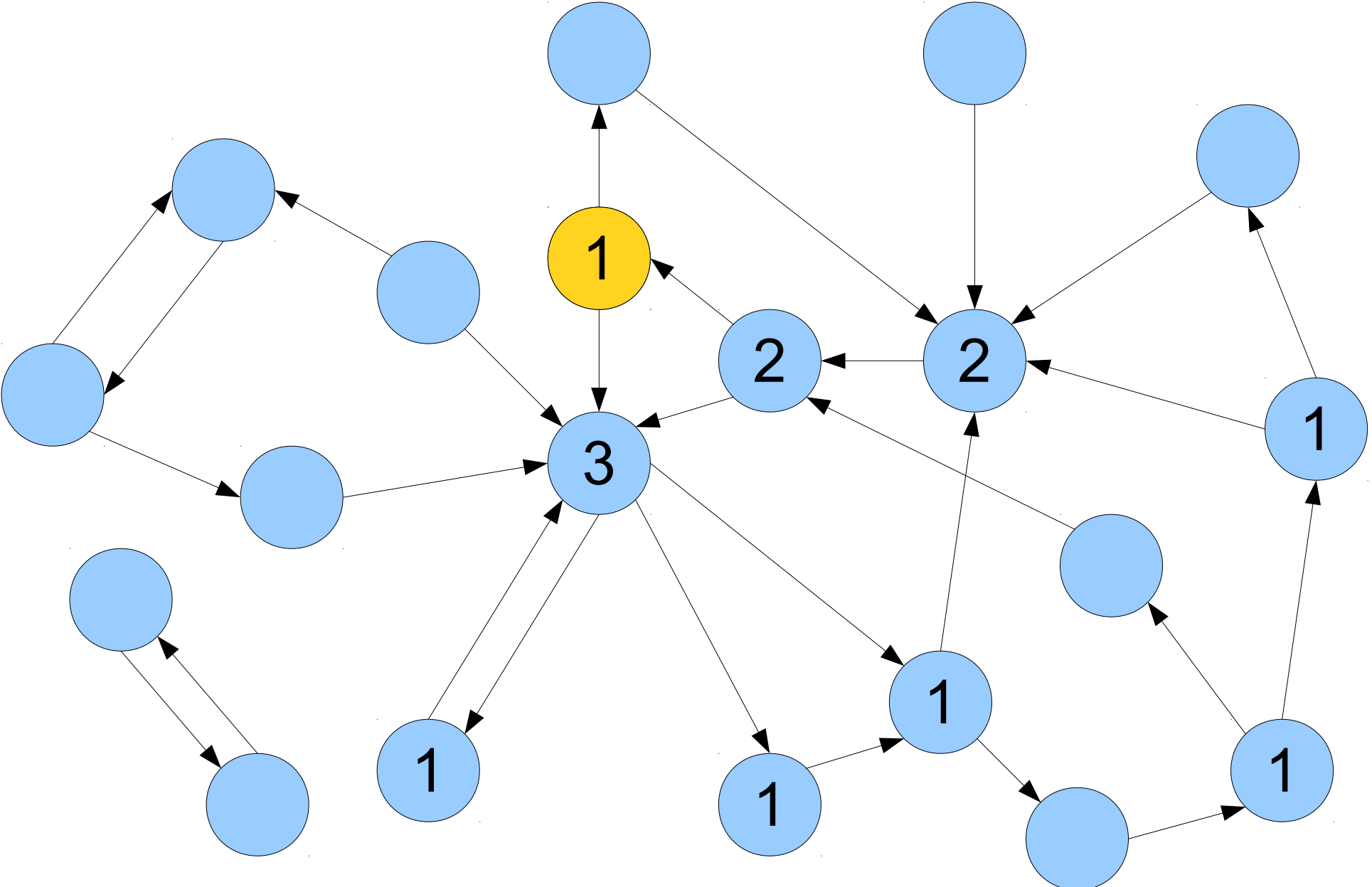
The Random Surfer Model



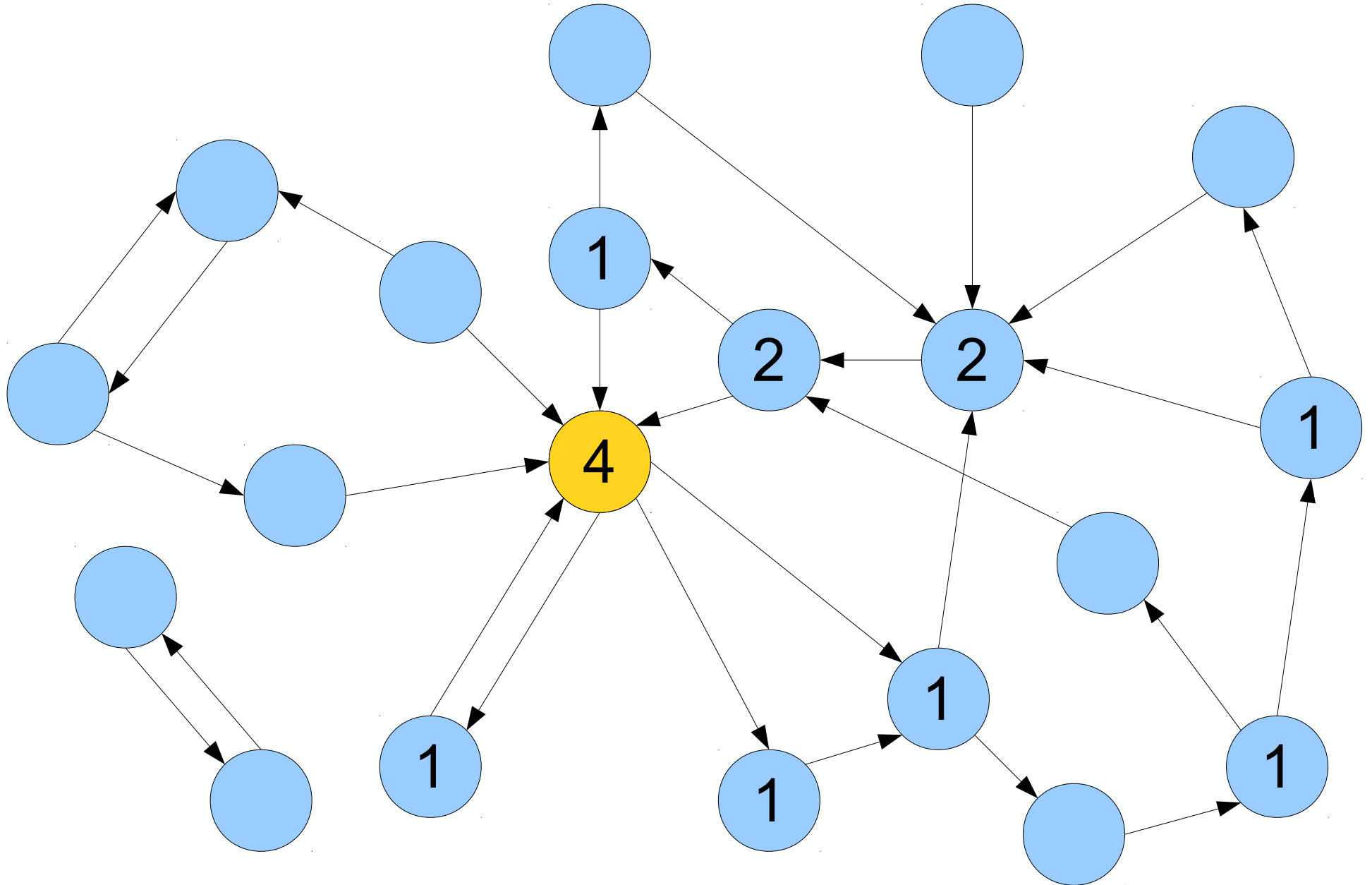
The Random Surfer Model



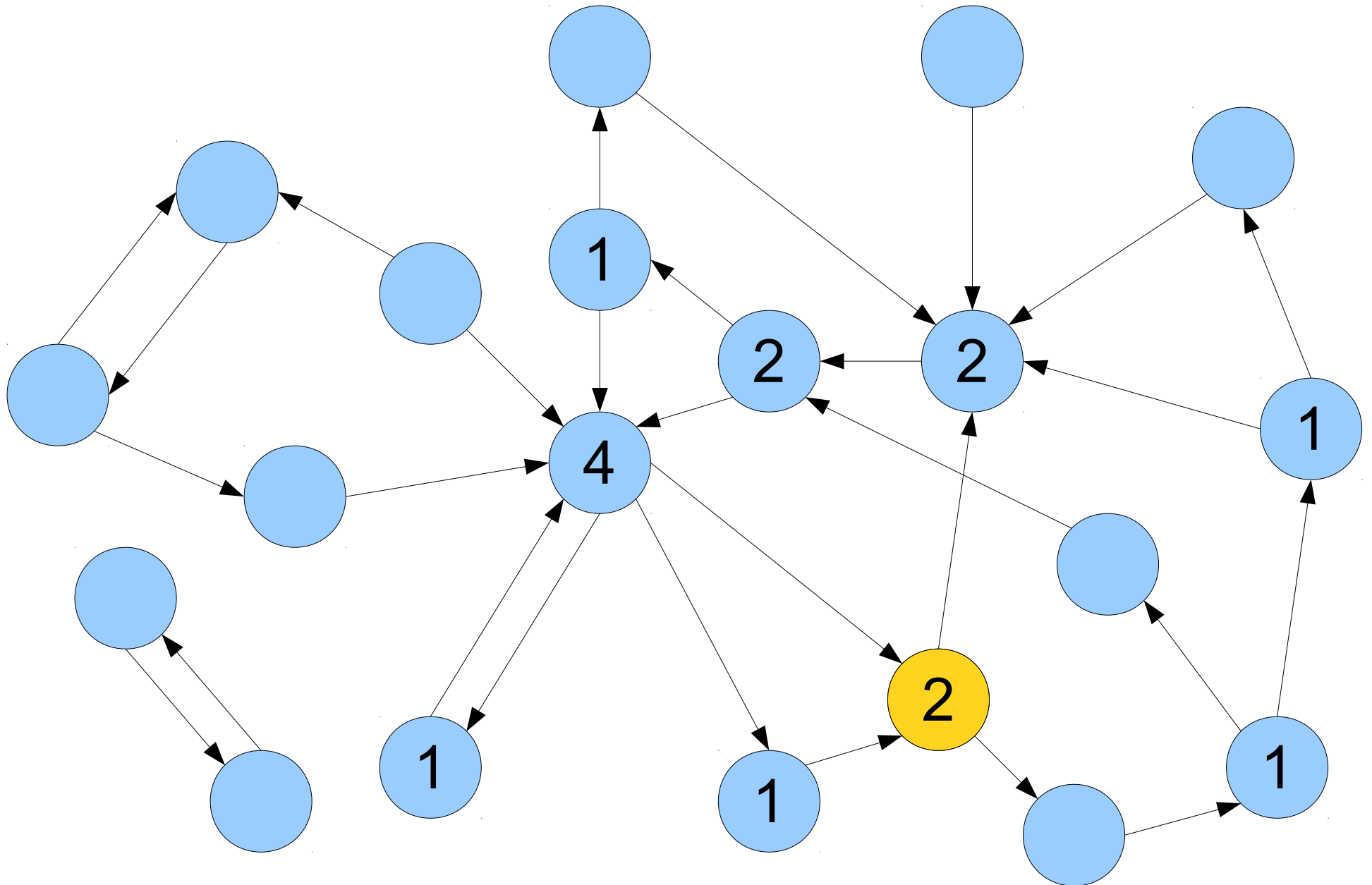
The Random Surfer Model



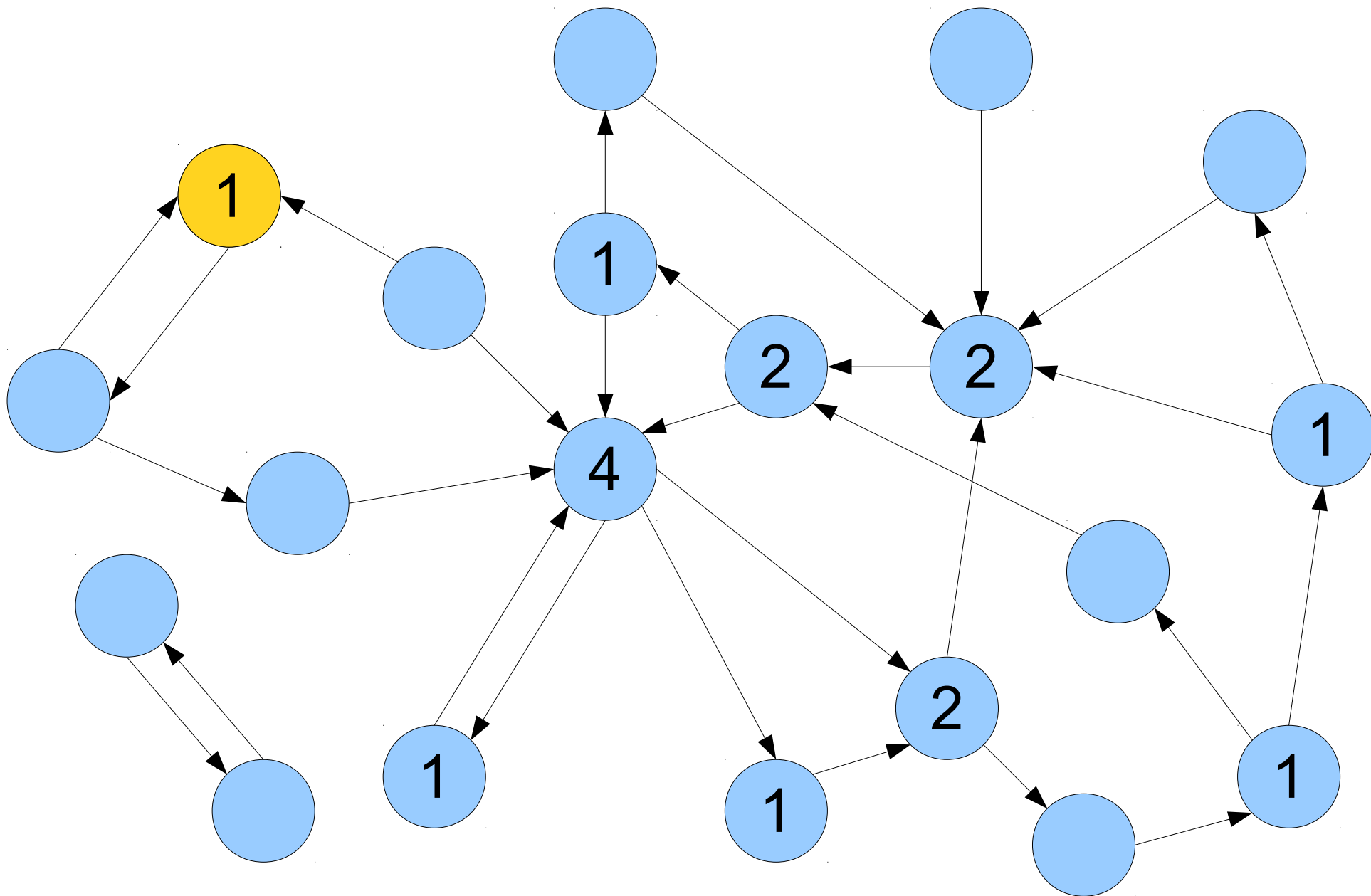
The Random Surfer Model



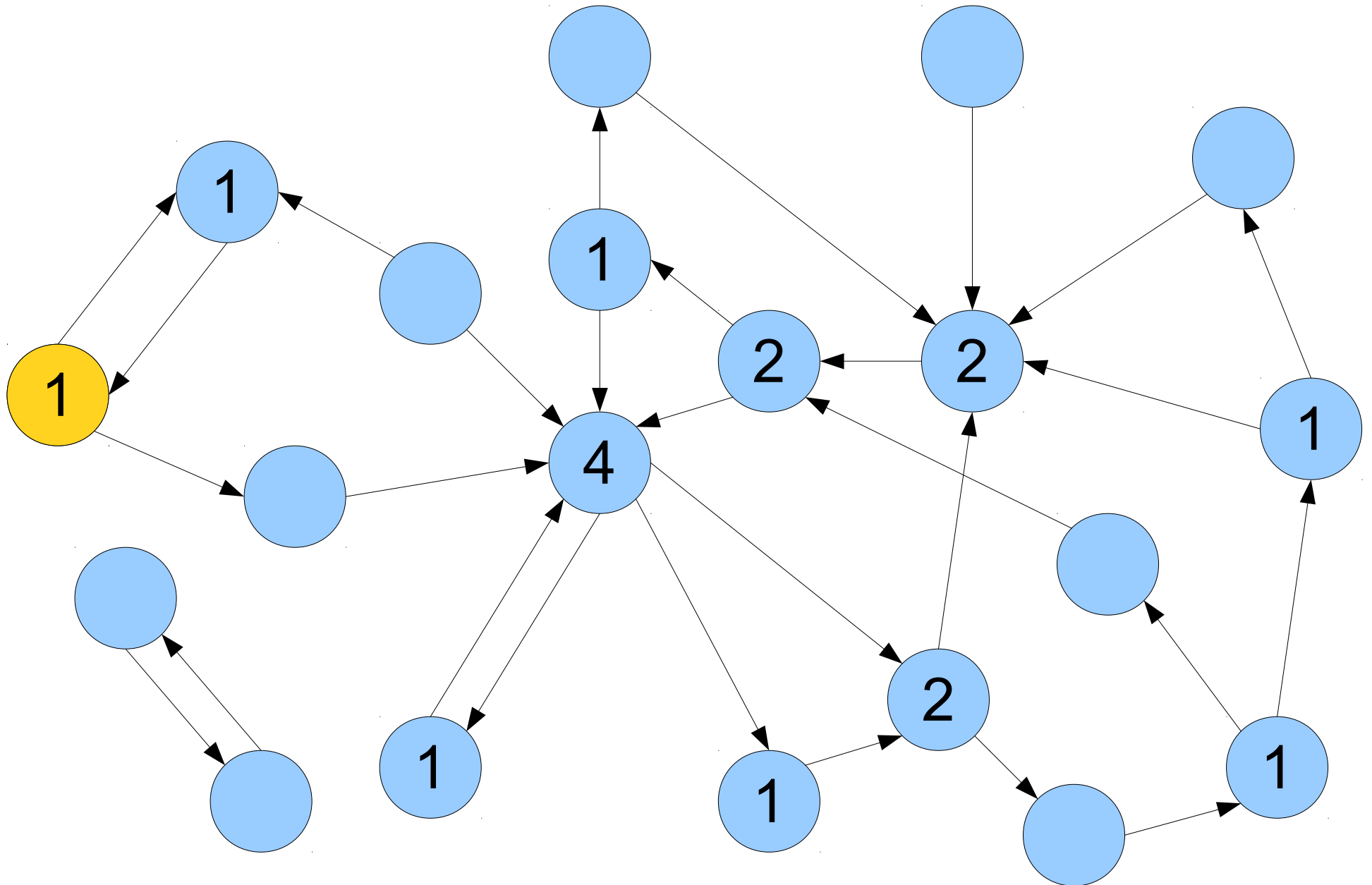
The Random Surfer Model



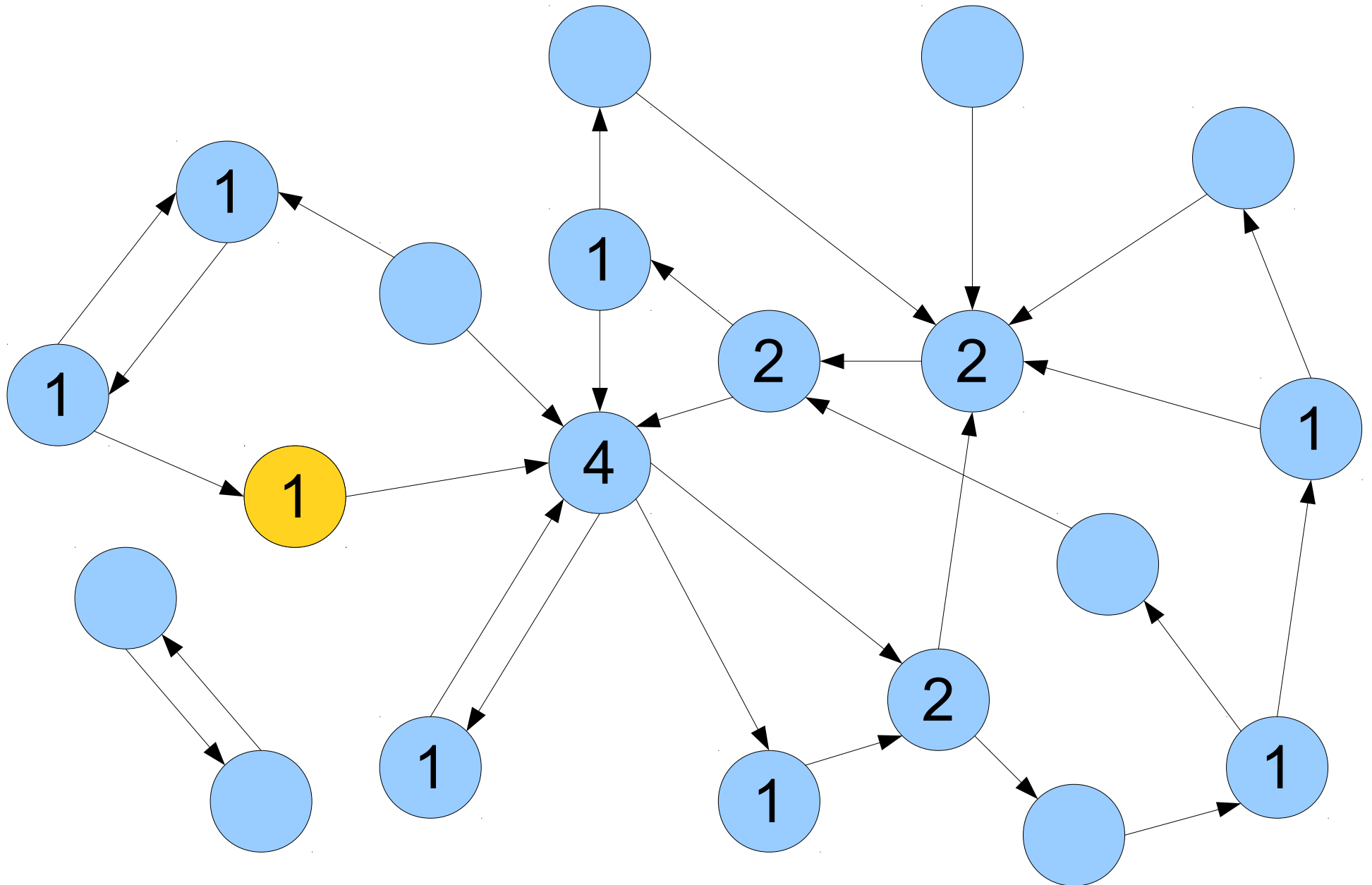
The Random Surfer Model



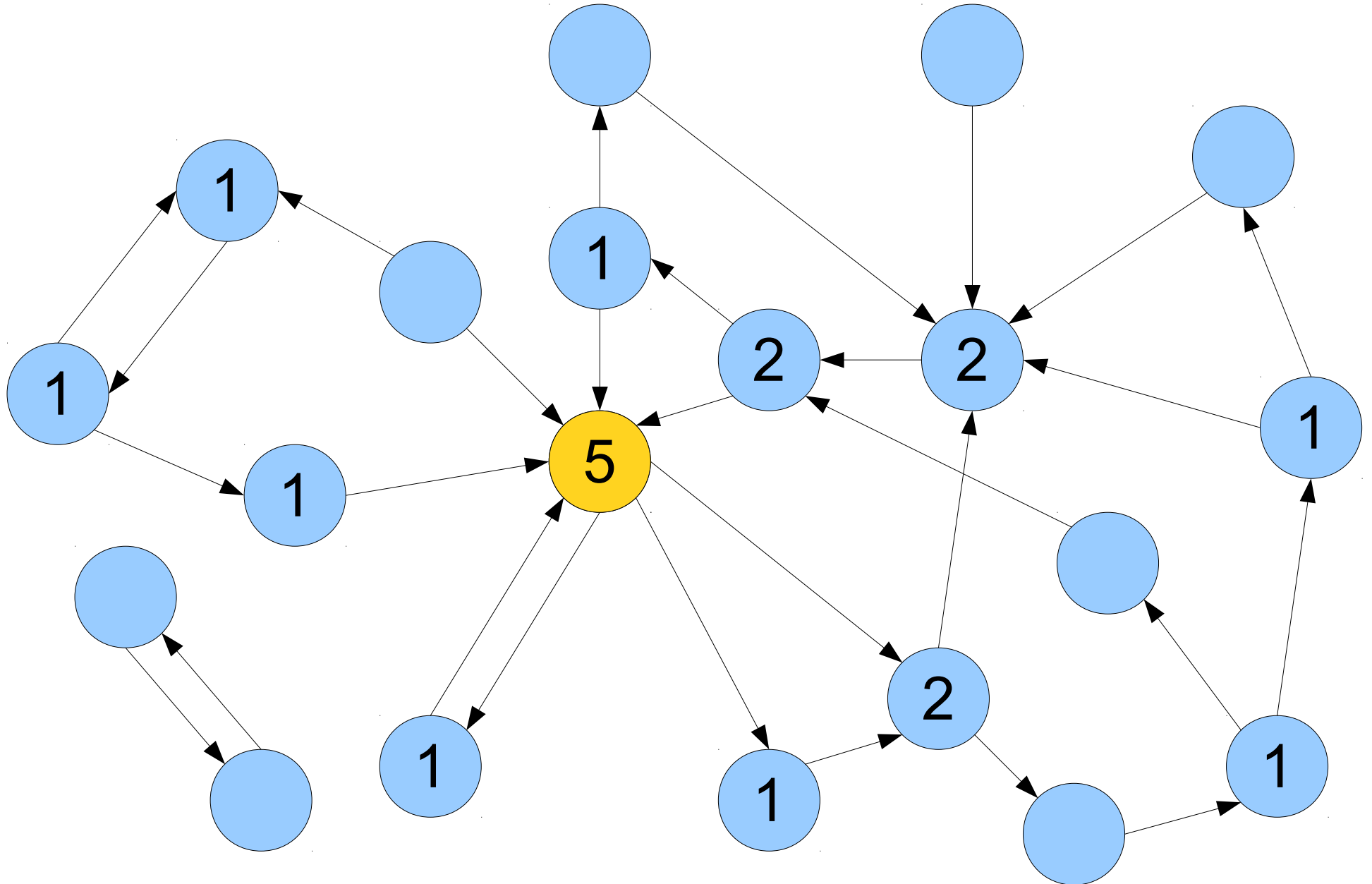
The Random Surfer Model



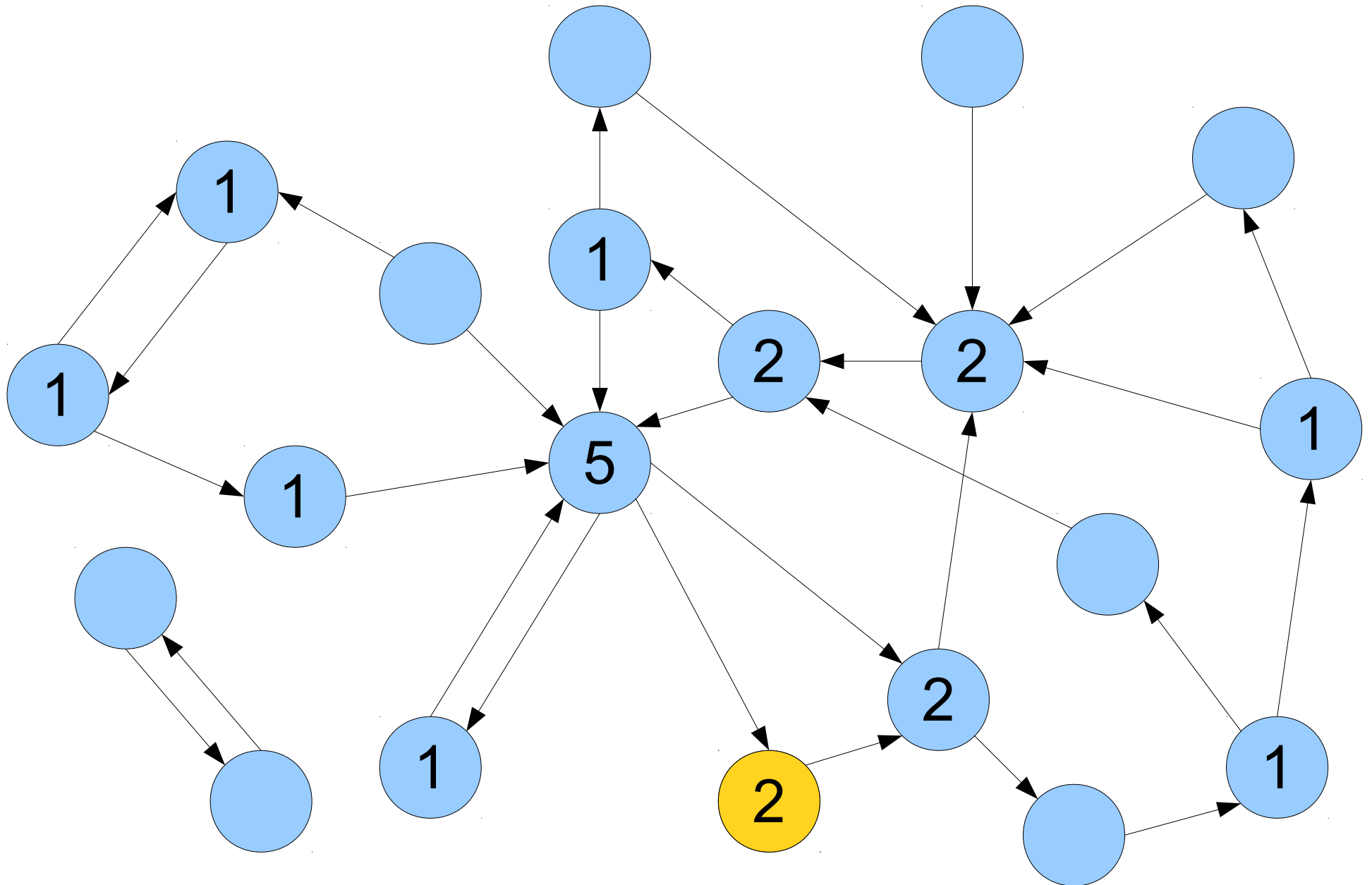
The Random Surfer Model



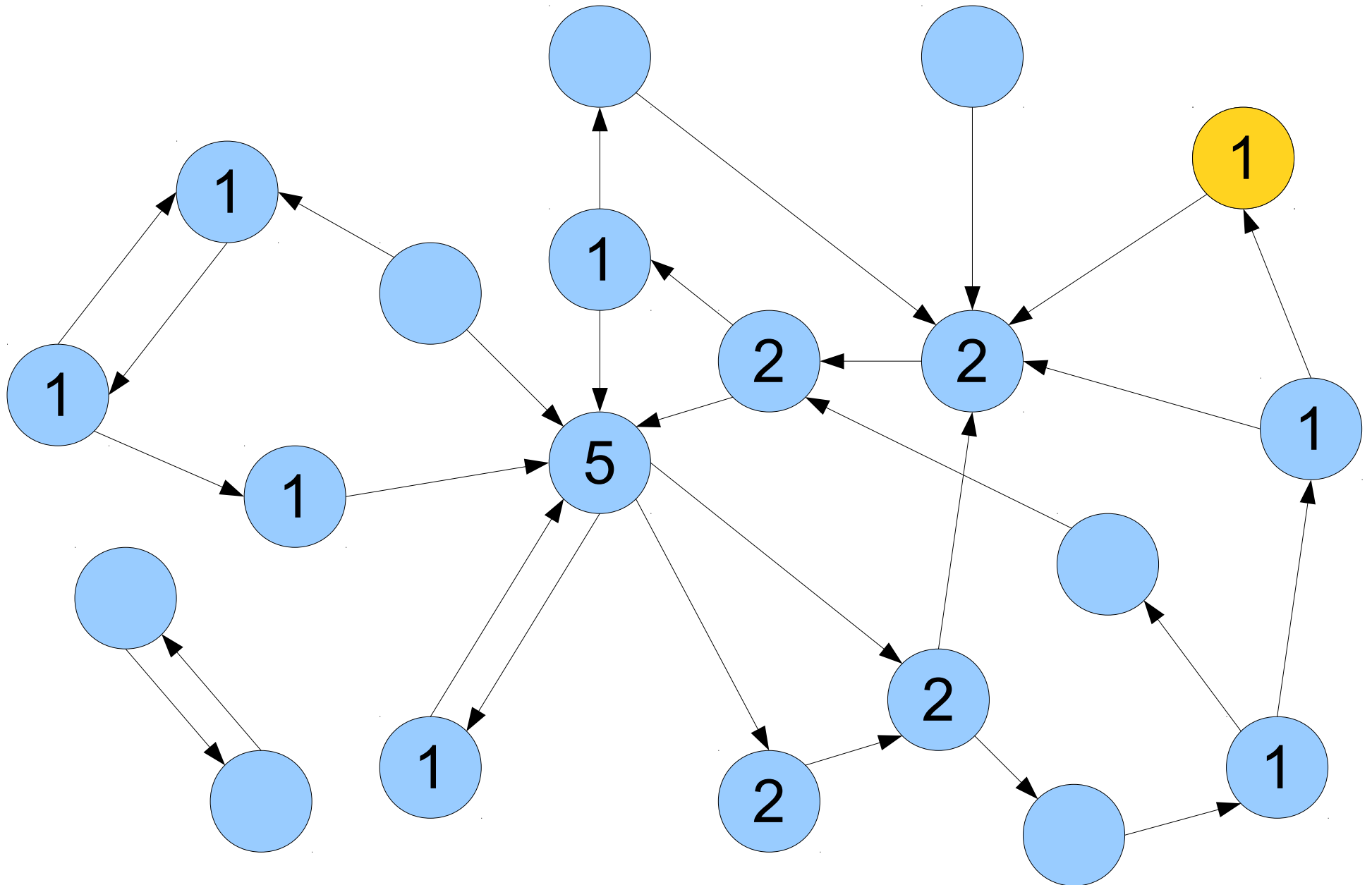
The Random Surfer Model



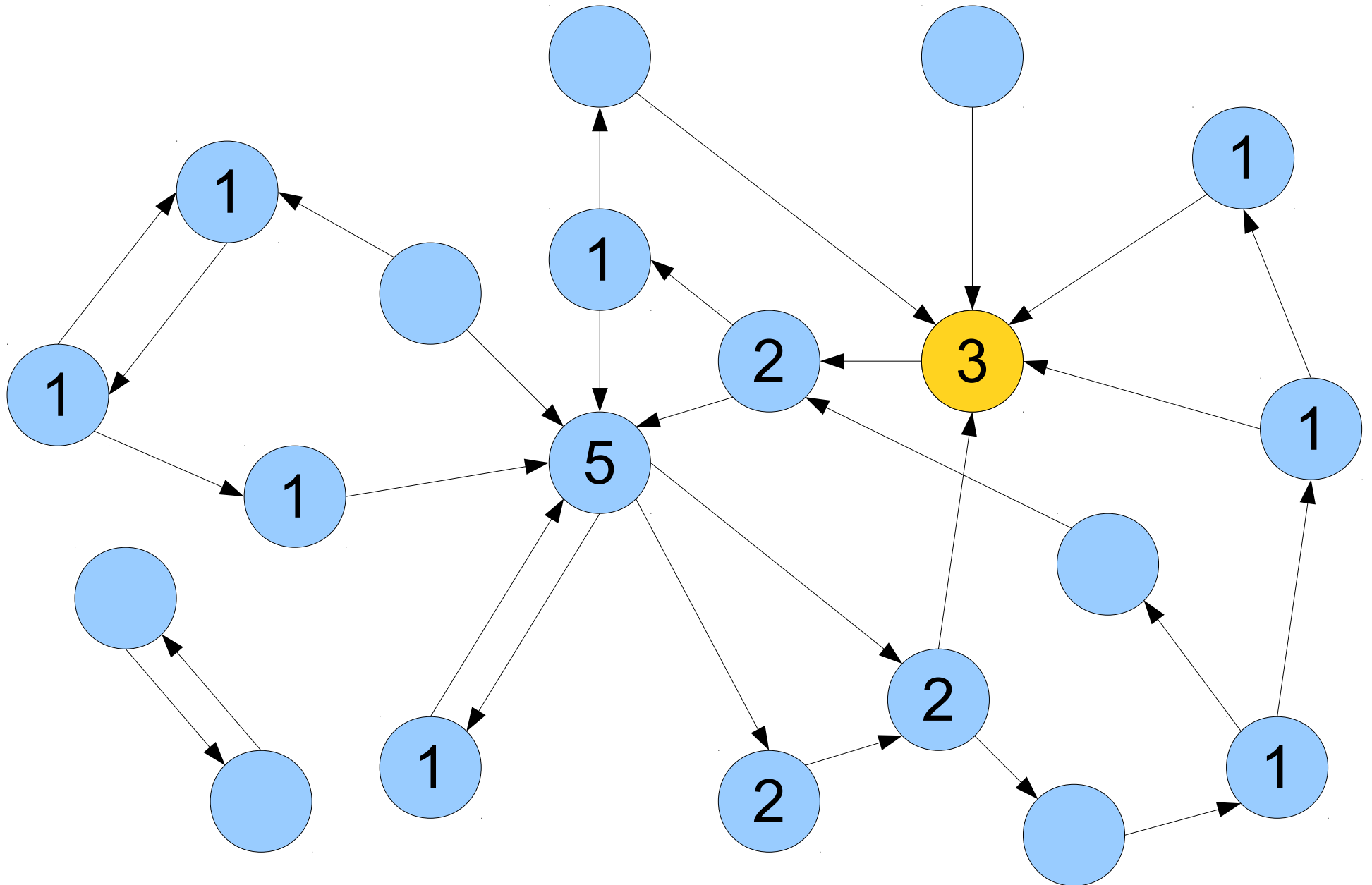
The Random Surfer Model



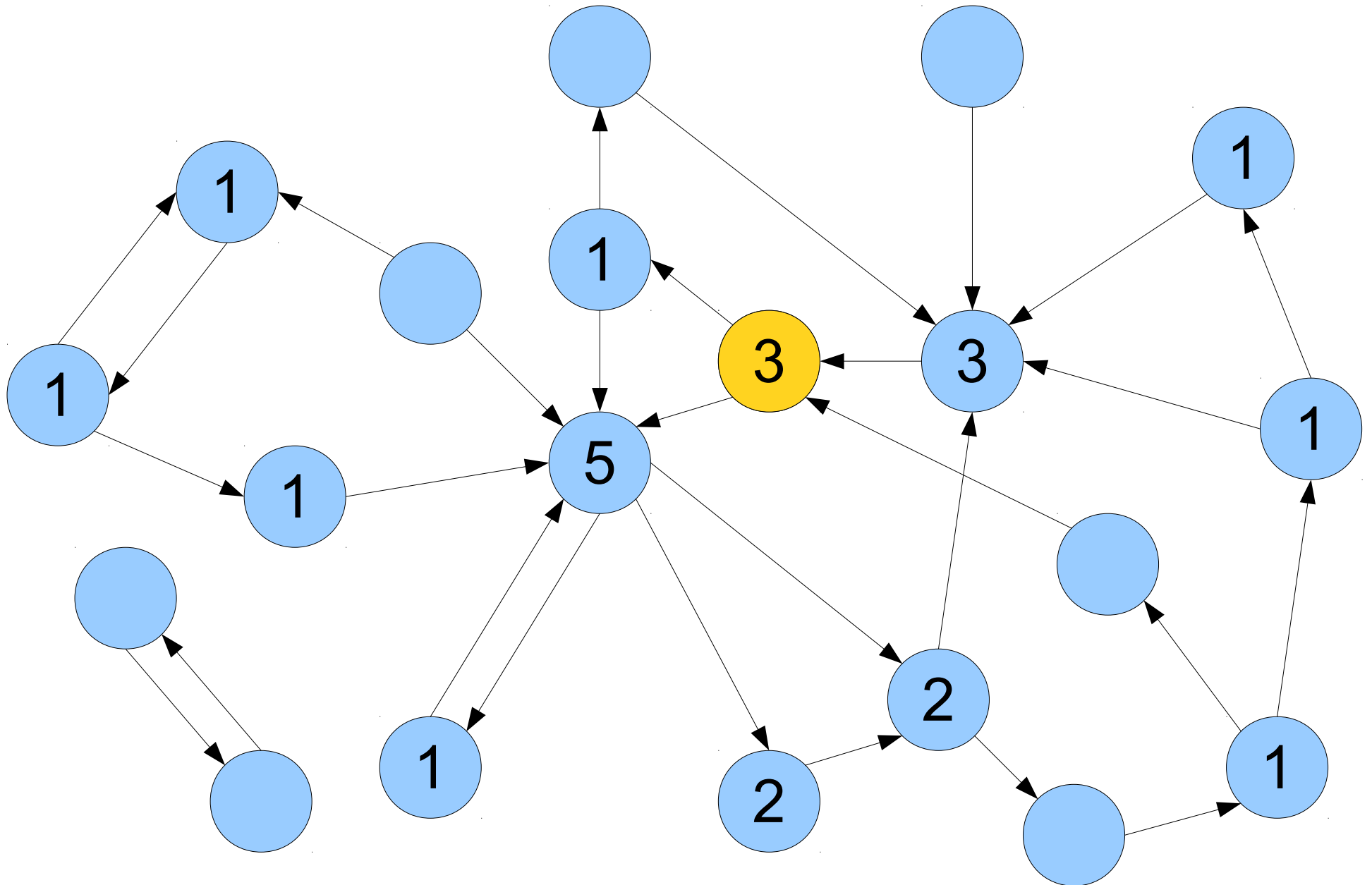
The Random Surfer Model



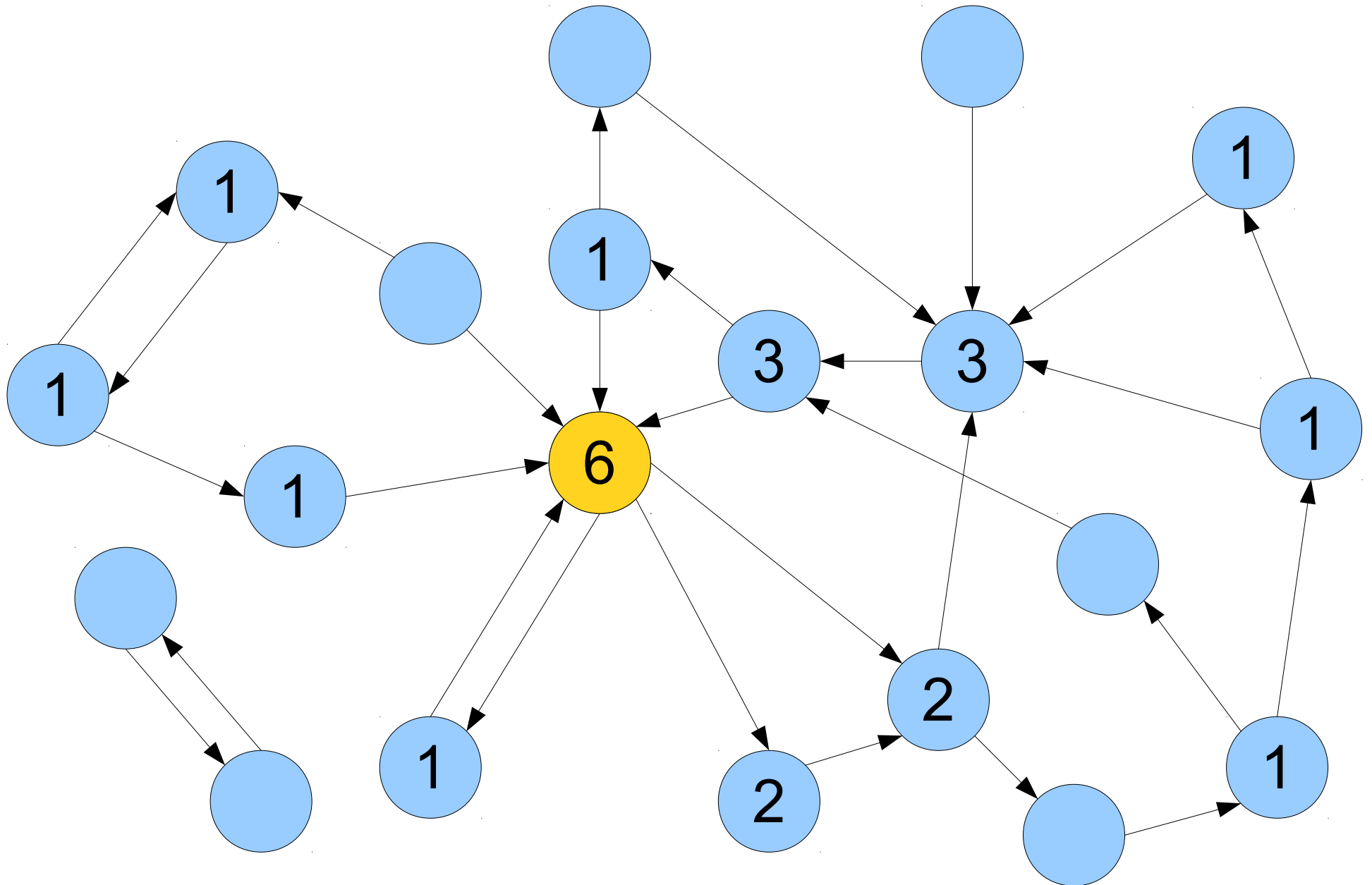
The Random Surfer Model



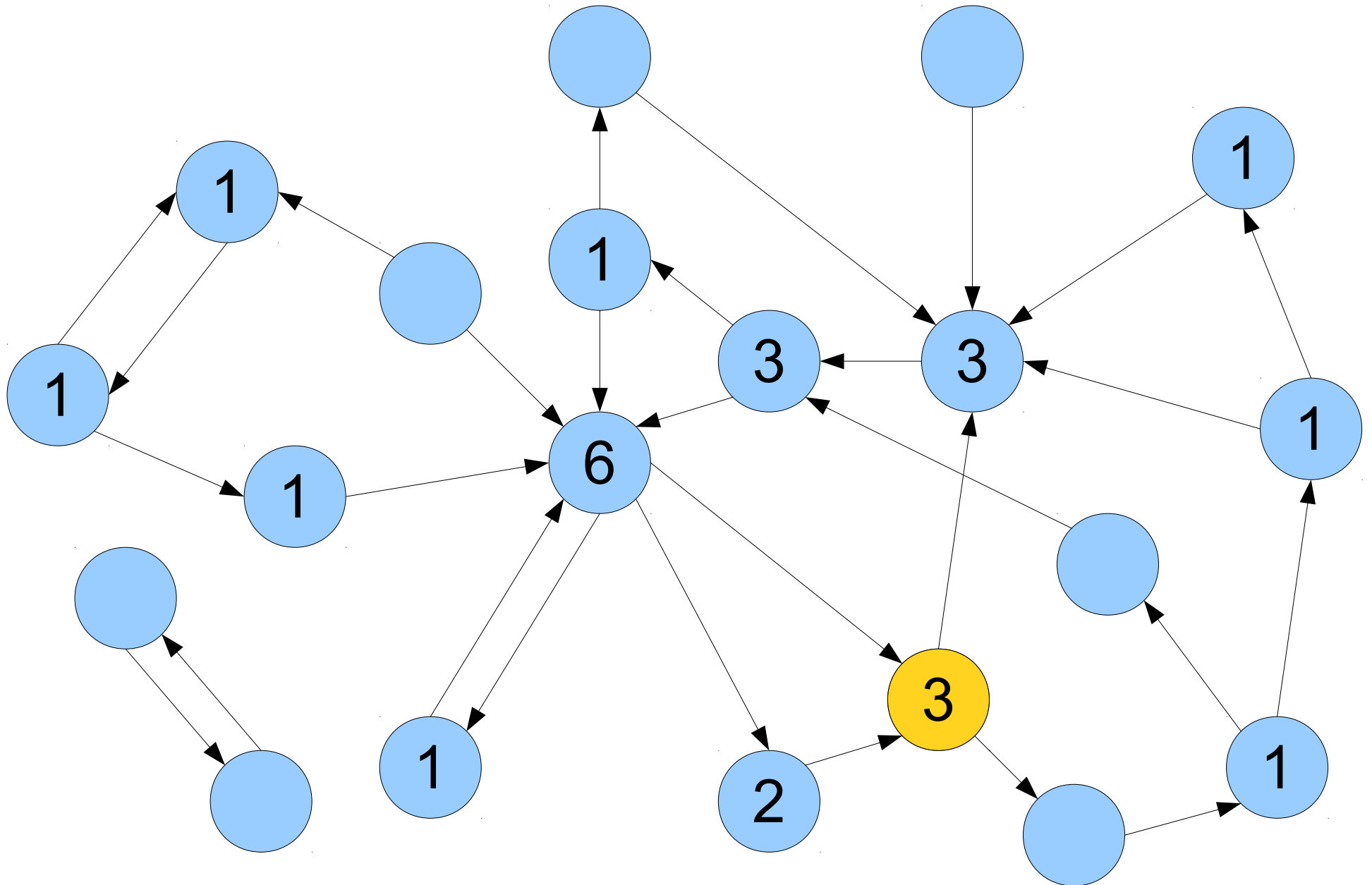
The Random Surfer Model



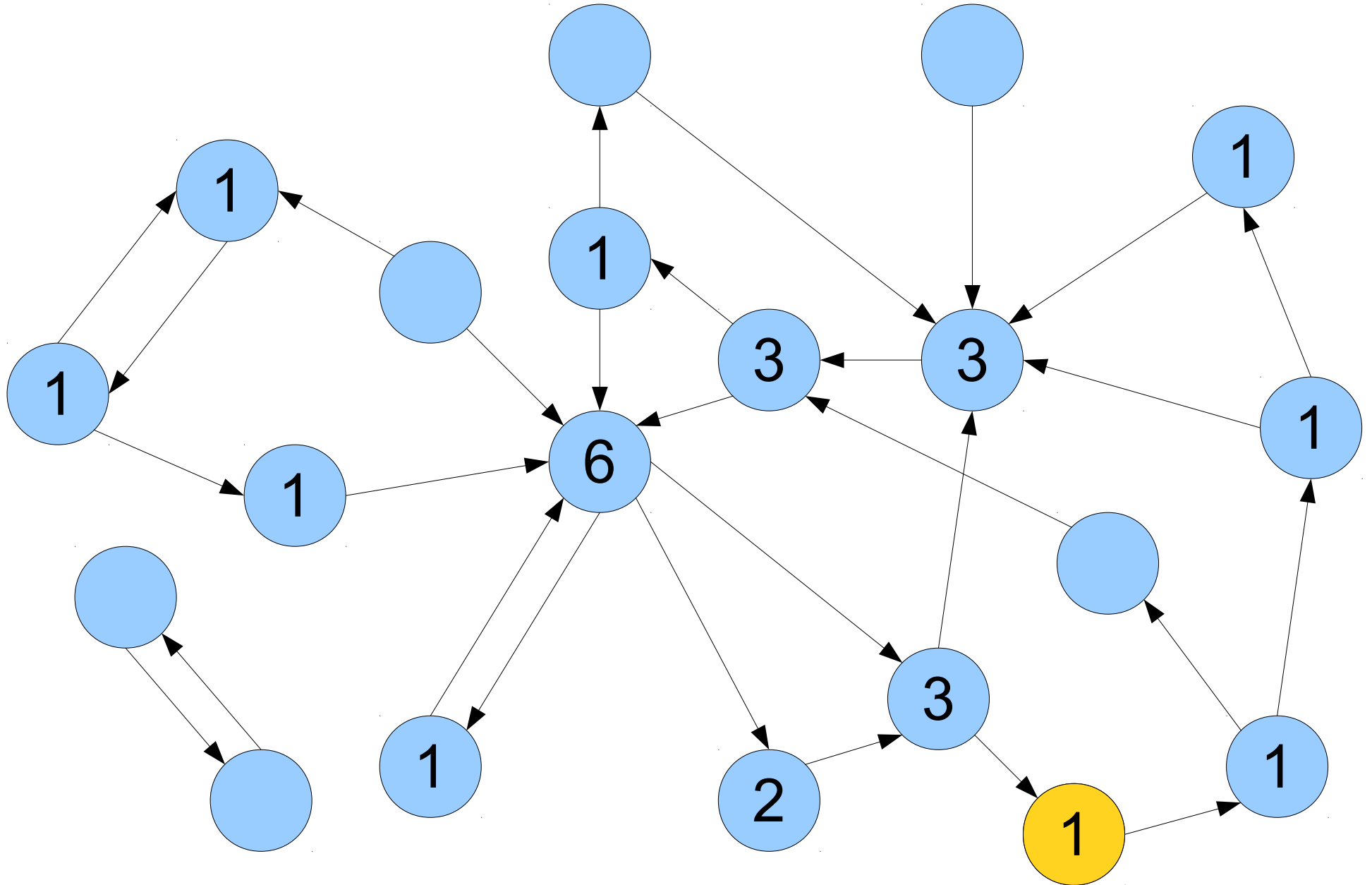
The Random Surfer Model



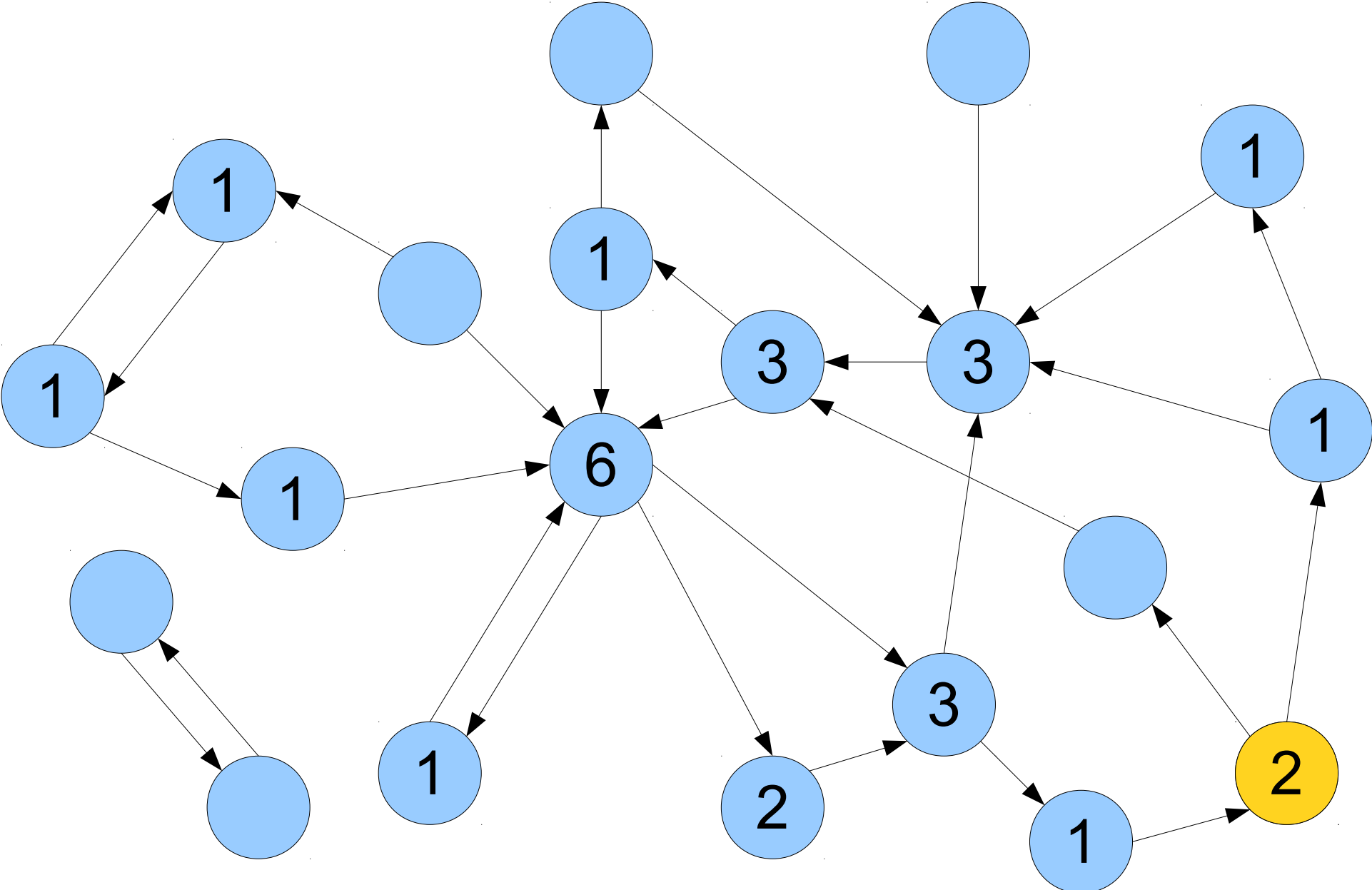
The Random Surfer Model



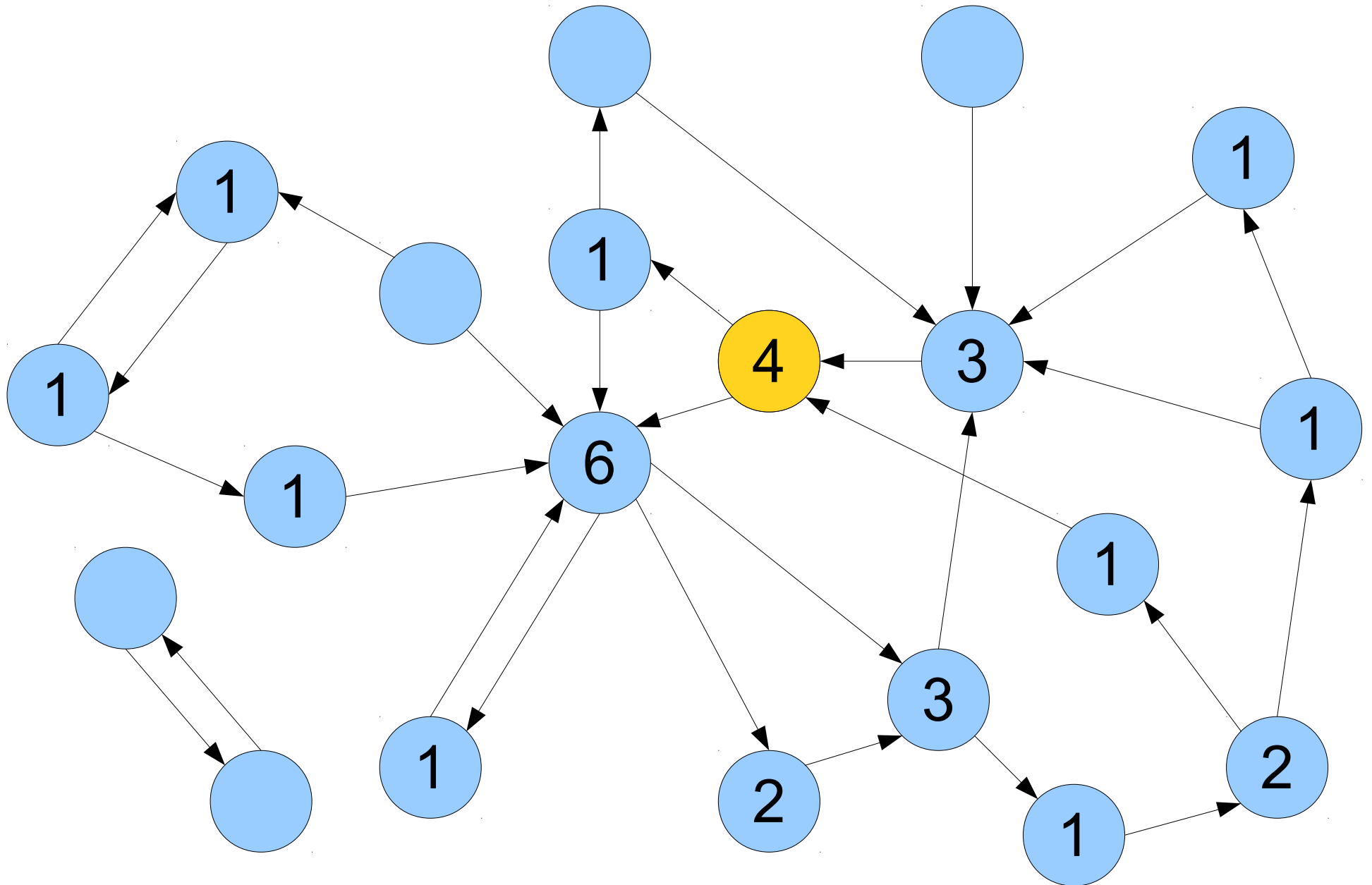
The Random Surfer Model



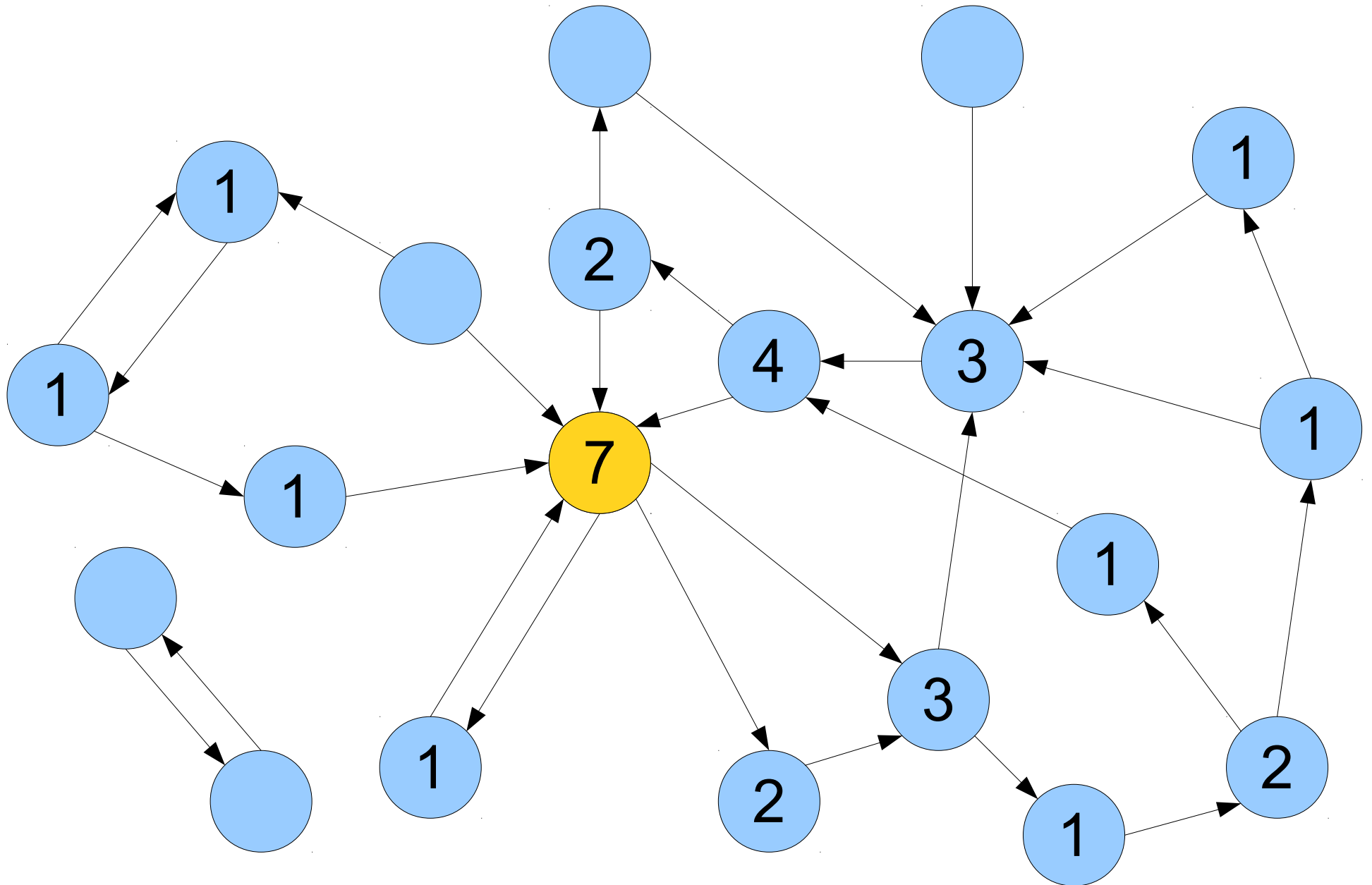
The Random Surfer Model



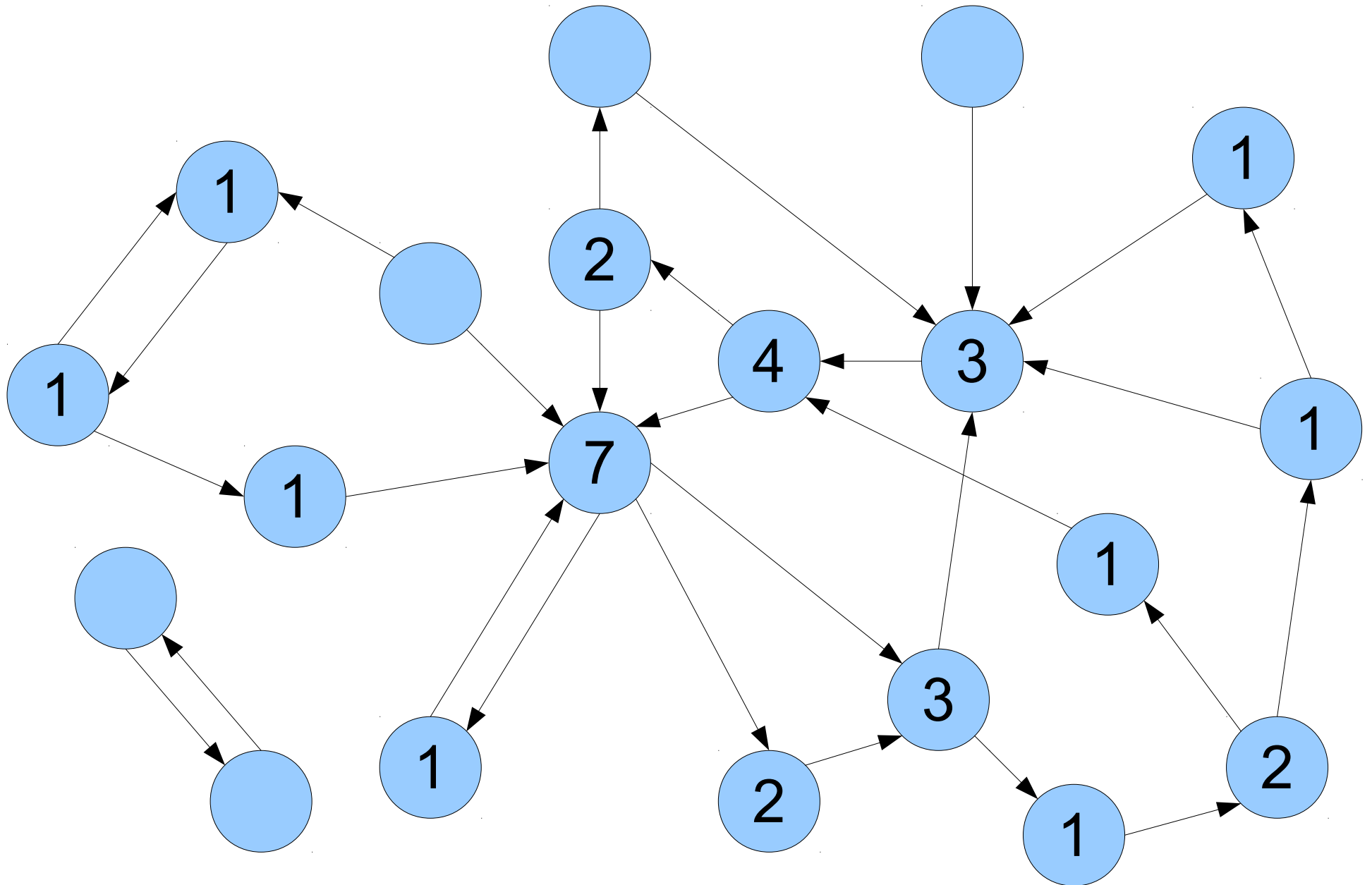
The Random Surfer Model



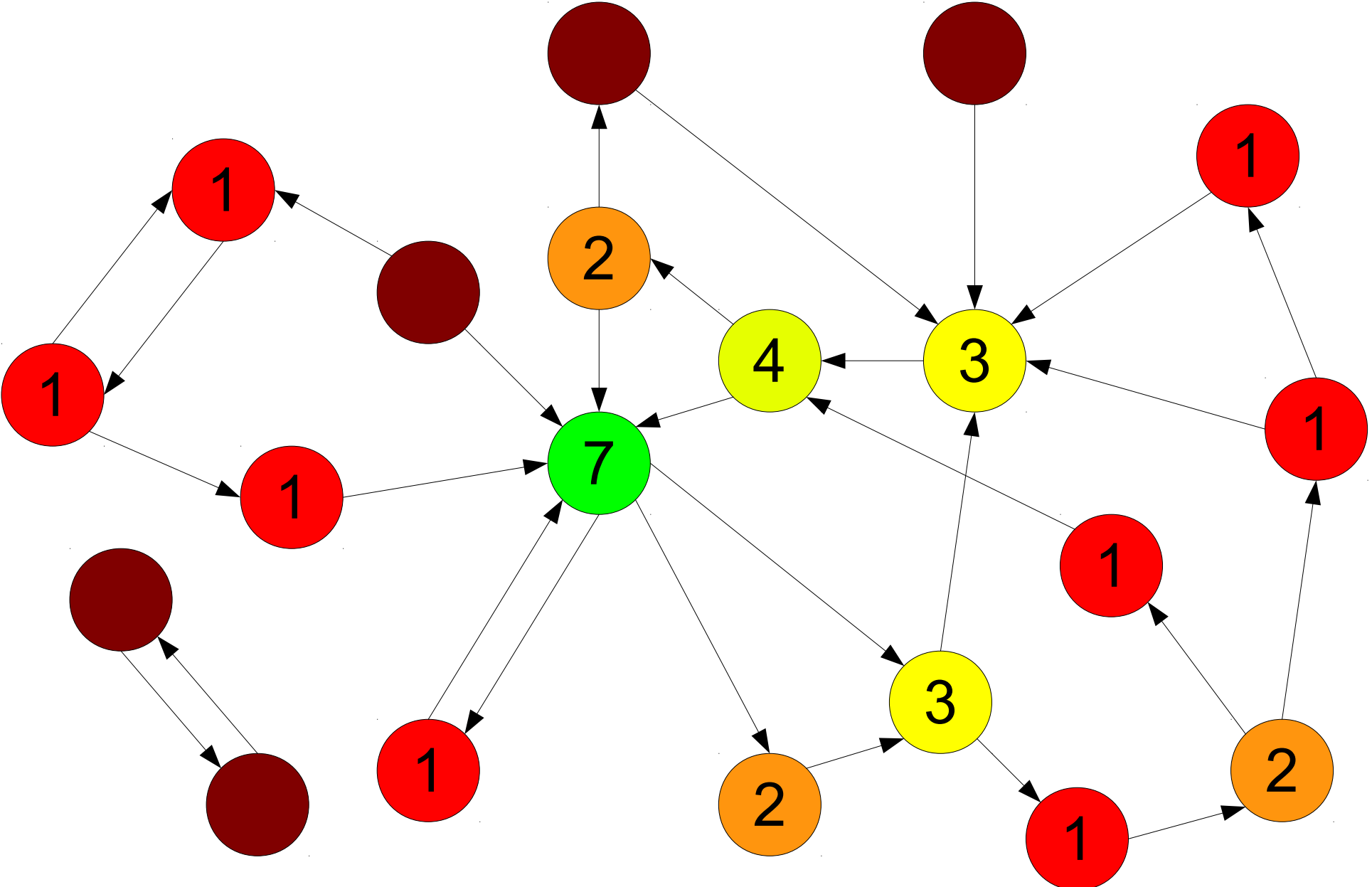
The Random Surfer Model



The Random Surfer Model



The Random Surfer Model



Ranking Articles with the RSM

- Randomly walk through the graph.
- At each step, either
 - Jump to a totally random article, or
 - Follow a random link.
- Record how many times each article was visited.
- The most-visited articles are, in some sense, the most important.

Other Applications of the RSM

- Ecosystem Stability:
 - Represent all the species in an ecosystem as nodes.
 - Draw edges representing energy flow through the ecosystem (predator/prey).
 - “Important” nodes represent places where energy concentrates; important for extinction models.
- Learn more:
 - <http://news.bbc.co.uk/2/hi/8238462.stm>

Who invented this?

[Our approach] can be thought of as a model of user behavior. We assume there is a "random surfer" who is given a web page at random and keeps clicking on links, never hitting "back" but eventually gets bored and starts on another random page.

[Our approach] can be thought of as a model of user behavior. We assume there is a "random surfer" who is given a web page at random and keeps clicking on links, never hitting "back" but eventually gets bored and starts on another random page. The probability that the random surfer visits a page is its **PageRank**.

The Anatomy of a Large-Scale Hypertextual Web Search Engine

Sergey Brin and Lawrence Page
Computer Science Department,
Stanford University, Stanford, CA 94305, USA
sergey@cs.stanford.edu and page@cs.stanford.edu

