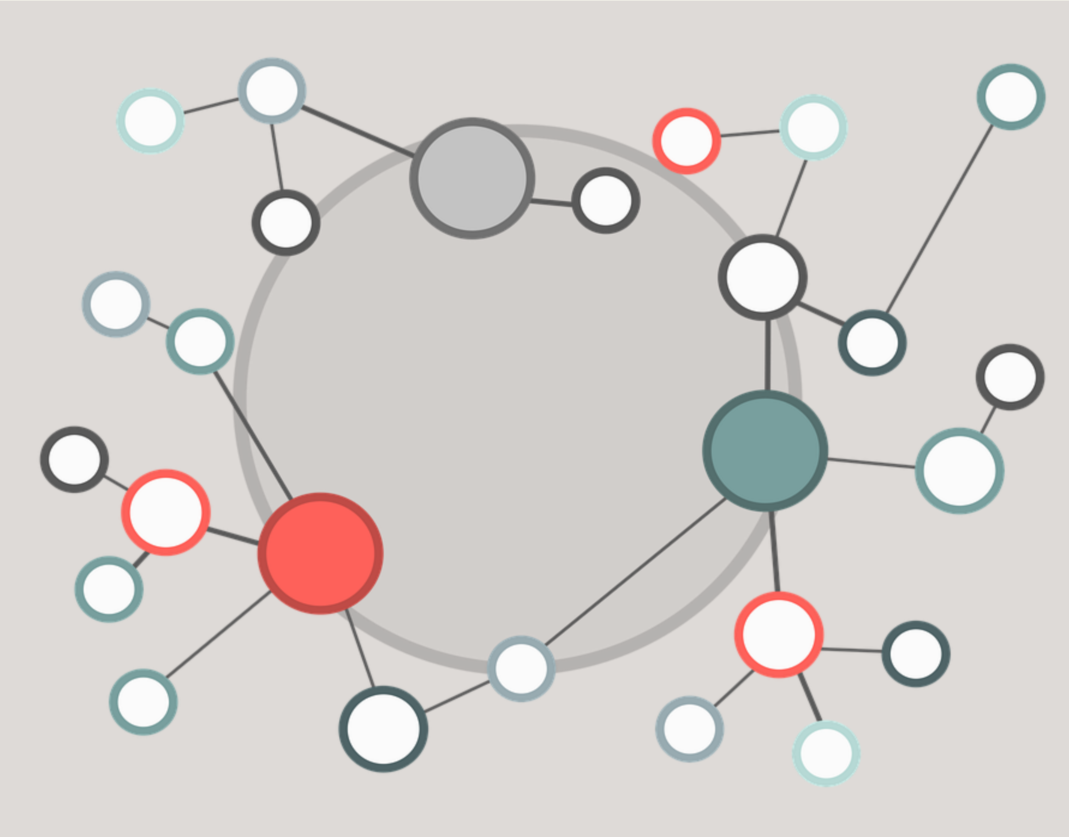
BLOCKCHAIN P2P NETWORK

Blockchain technologies, **lecture 4**



Course overview

- Block and transaction propagation
- Topology
- Discovery protocol
- Broadcasting, messages

- Kademlia -- peer to peer information system, RLPx
- Wire protocol

BITCOIN TOPOLOGY

Bitcoin topology

- Nodes:
  - validate transactions;
  - propagate transactions;
  - propagate blocks;
  - discover new peers;
- Self-organized network.
- Self-configured.

Bitcoin topology

- Nodes in the network form a random graph.
- Newly joined nodes query DNS servers.
- DNS servers return a random set of bootstrap nodes.
- A node learns about other nodes by listening from advertisements of new addresses coming from their neighbors.
- Each node keeps list of opened connections. Node randomly selects an address from a set of known addresses and attempts to establish a connection.
- Default number of connections: 8. Node’s number of connections may exceed the default number due to incoming connections.

Bitcoin topology

- Each node tries to connect to peers using TCP (*outbound connections*).
- Default number of outbound connections: 8.
- A node stores IP addresses in two lists: **new** and **tried**.
    - **new list**. Addresses of peers to which the node has not yet tried to connect.
    - **tried** list. Addresses known as reachable
- A nodes accepts *inbound connections* from other peers.

Bitcoin topology

- A nodes accepts *inbound connections* from other peers.
- A listening node (a node that accepts inbound connections) may issue

  ADDR messages to advertise neighbors that it accepts inbound connections

  neighbors may relay ADDR message to their own neighbors by following

  a *gossip protocol*.

- A node may request to discover other active peers by sending a

  GETADDR message.

- A nodes periodically verifies the state of the nodes it is connected to by issuing a

  PING messages and waiting for PONG responses.

Bitcoin topology

- **Full nodes** download every block and every transaction and verifies all consensus rules.
  - store: UTXO as a LevelDB database;
  - block index: A Leveldb database containing meta-data about all blocks;
  - block raw Blocks in raw format;
  - undo data Data used when rolling back the chain-state, in case of reorganization of the chain.

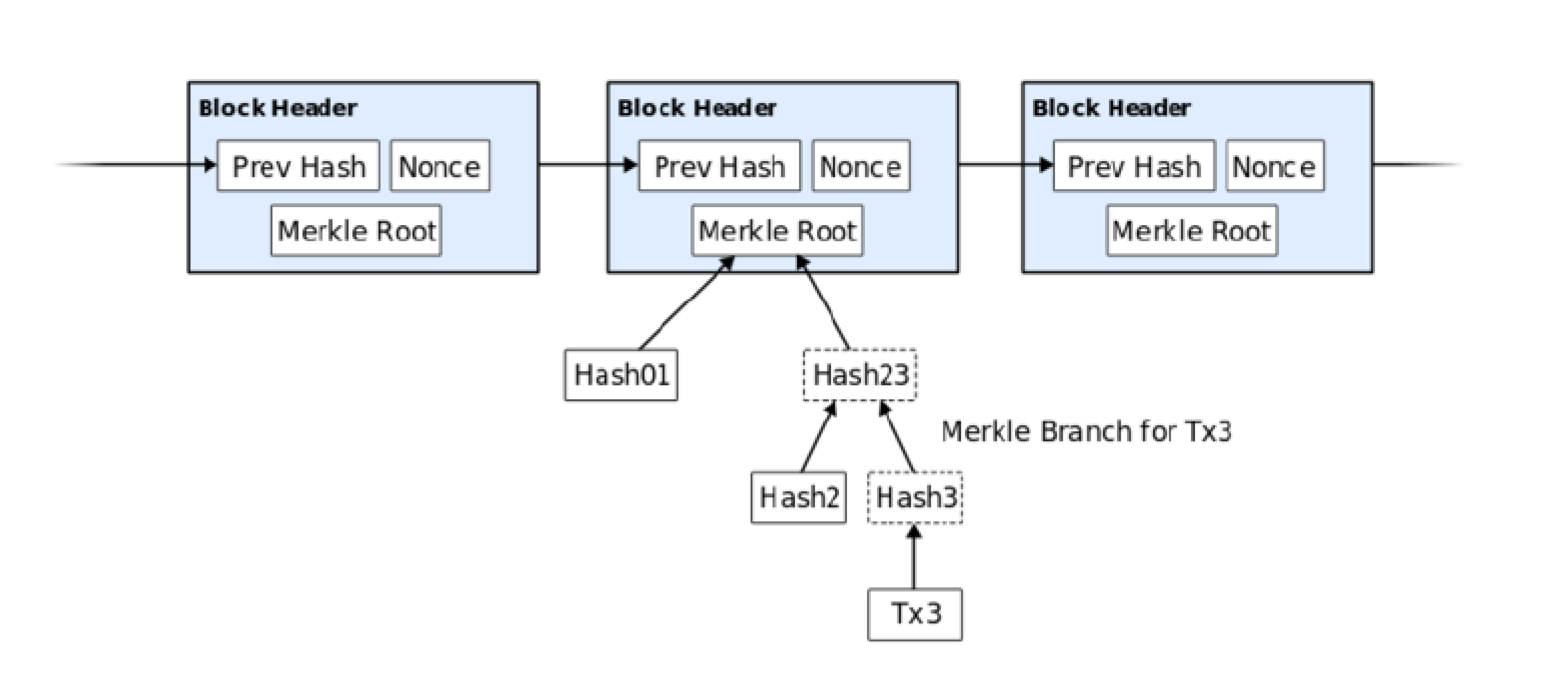
- **Miners** nodes extending the blockchain by creating new blocks.
  - Miners may work alone or in mining pool with an administrator running a full node.
  - A mining pool distributes rewards based on each individual's contribution to the processing power for the group

Bitcoin topology

- **Lightweight nodes** download only **block headers** and relies on full nodes for the full blockchain ledger.
  - Full nodes serve lightweight clients by notifying them when a transaction affects their wallet (SPV wallet) and transmitting transactions to the network.
  - **Simplified Payment Verification,** prove that the sender is indeed the owner of the UTXOs sources of the payment, without downloading the full block chain.
  - SVP does not prove that funds have not been previously spent.

Bitcoin topology

- Lightweight nodes



Bitcoin topology

- **DNS seeder** server that responds to DNS query by initiating a message that contains a list of IPS. DNS seeds periodically crawl the network to obtain active IP addresses
    - DNS seeders are queried by new nodes
    - DNS seeders are queried by a node that restarts and tries to reconnect to new peers.

- DNS servers are hard-coded as trusted DNS servers maintained by the core developers.

- **SPAM score**. Each node scores peers, higher scores are assigned if a peer act as malicious node. Node stops sending messages to a peer that accumulated 100 points, for a period of 24h.

BITCOIN NETWORK MESSAGES

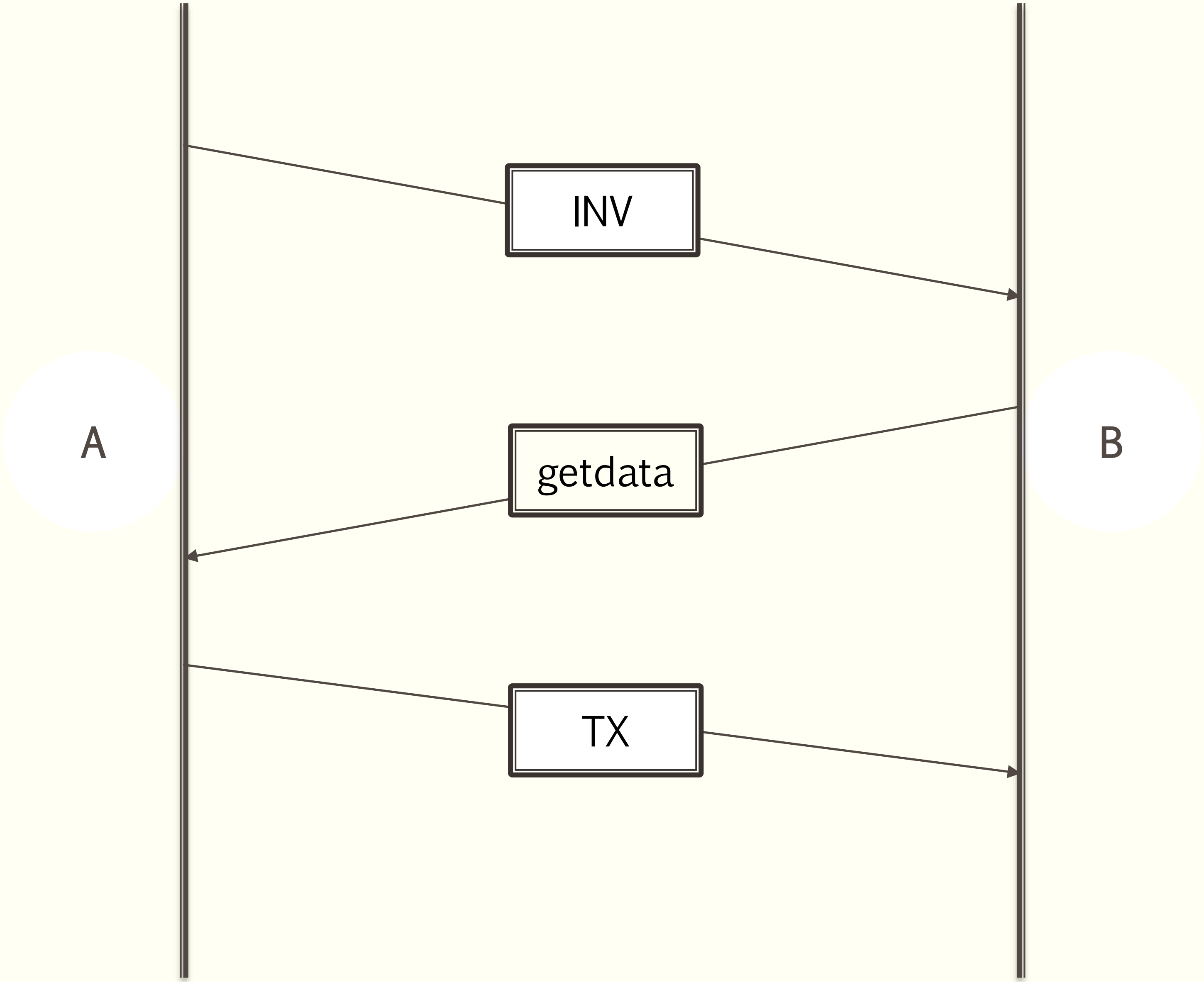
Updating and synchronization

- There are two types of transactions that update the distributed ledger replicas:

  tx messages and *block* messages.

- tx and block messages are advertised with inv messages.
  - **inv** message contains a set of transaction hashes and block headers received by the sender.
  - **getdata** message is issued by the receiver of inv message to the sender for a transaction or a block

- The propagation delay is the sum of transmission time and the local verification time of the block or transaction.


INV
A
getdata
B
TX

GOSSIP PROTOCOLS

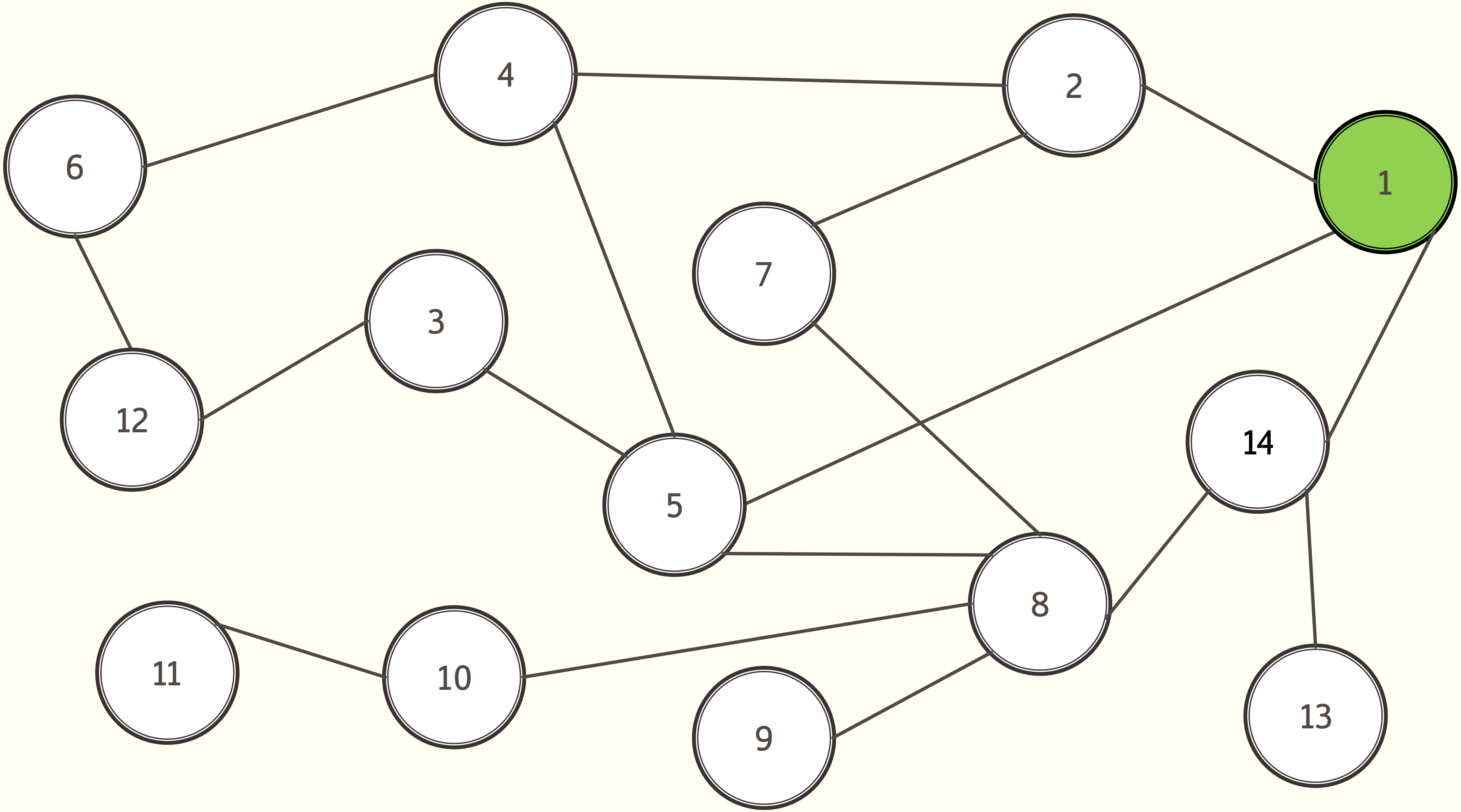
Gossip protocols

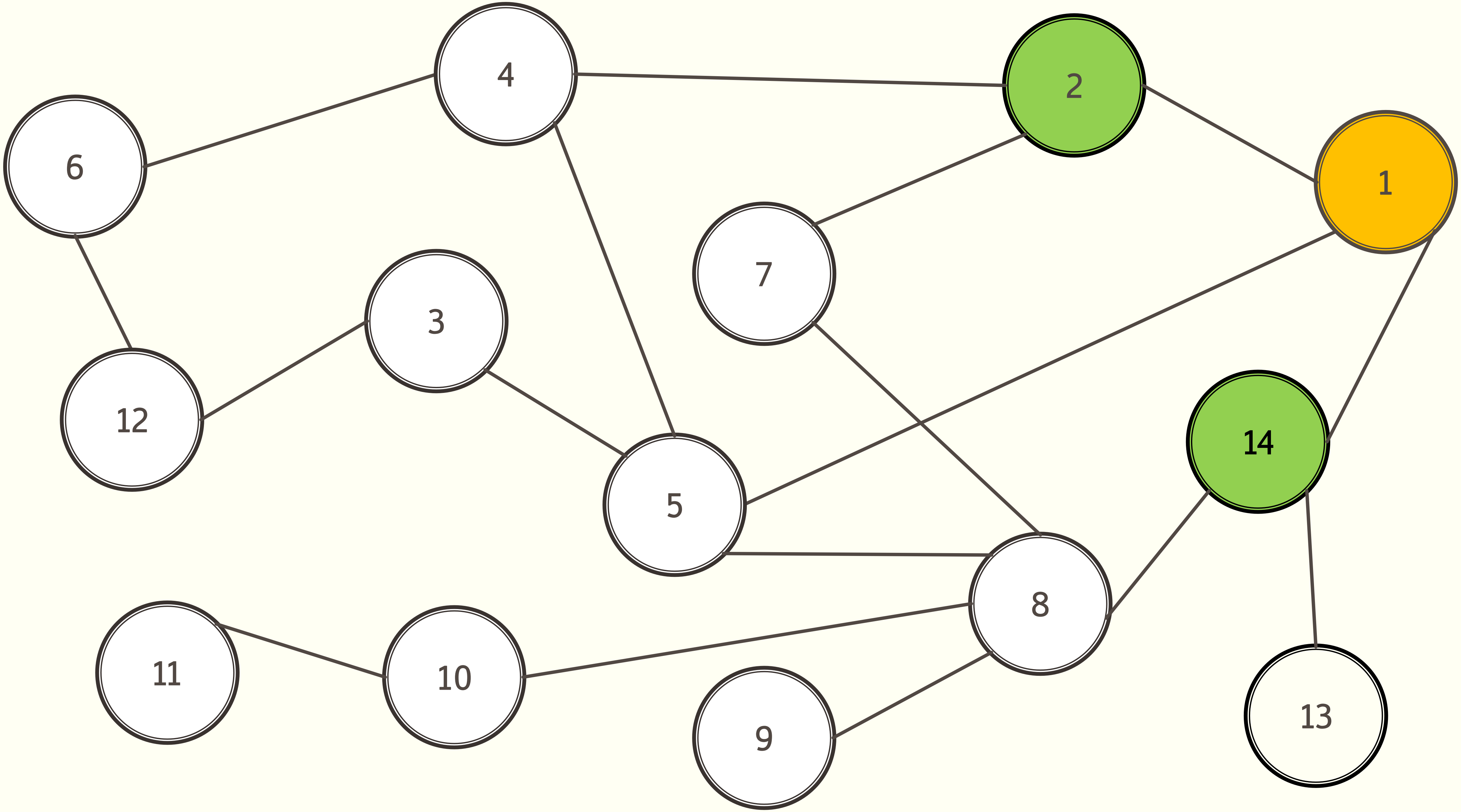
- Each node sends a message to K random targets (multicast).
- K -- infection factor.

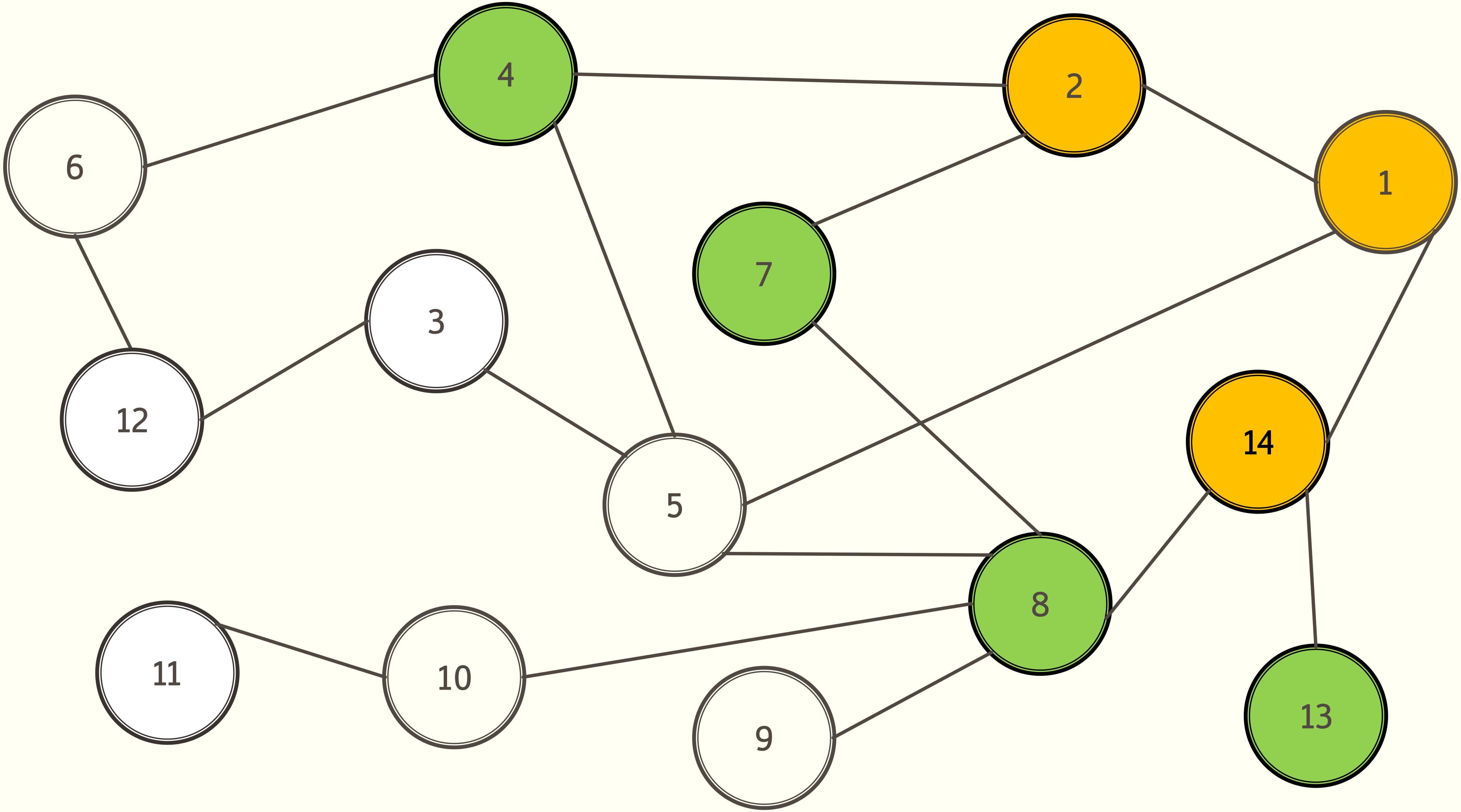
- Flooding: If a node gossips to all neighbors.

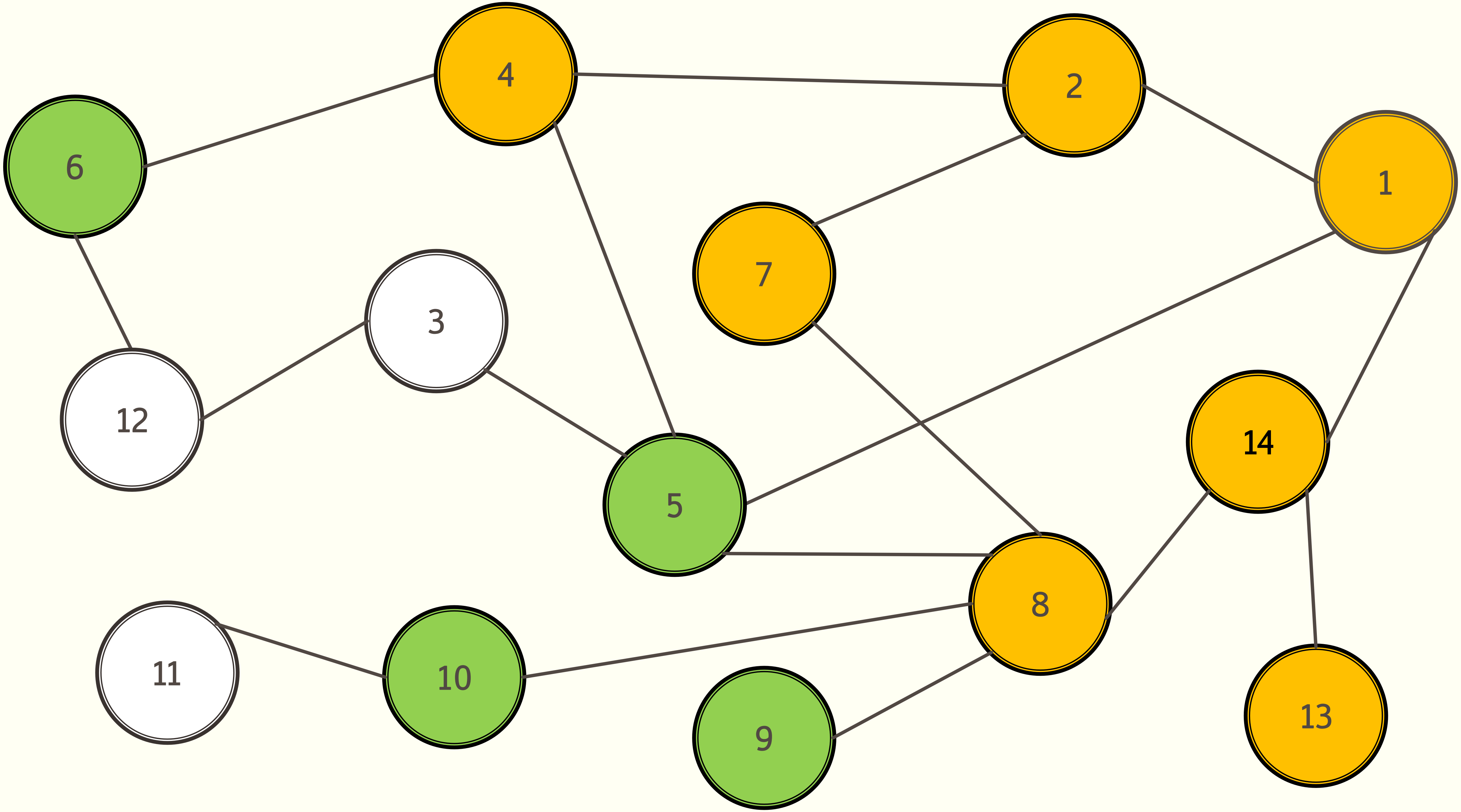
- Each target randomly select another K targets.

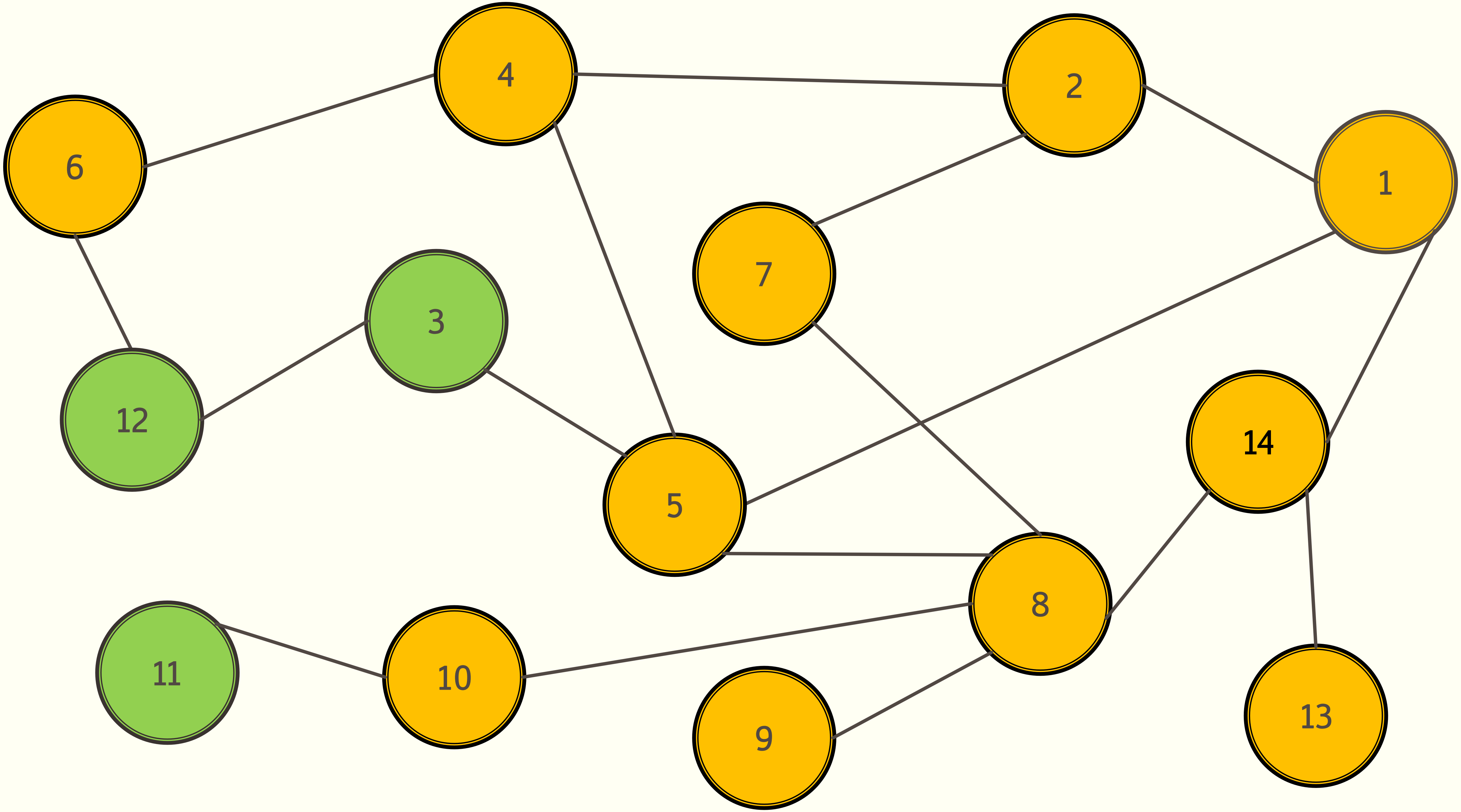
- Process stops when all nodes receive the message or when the message expires.


4
2
6
1
7
3
12
14
5
8
11
10
9
13


4
2
6
1
7
3
12
14
5
8
11
10
9
13


4
2
6
1
7
3
12
14
5
8
11
10
13
9


4
2
6
1
7
3
12
14
5
8
11
10
9
13


4
2
6
1
7
3
12
14
5
8
11
10
9
13

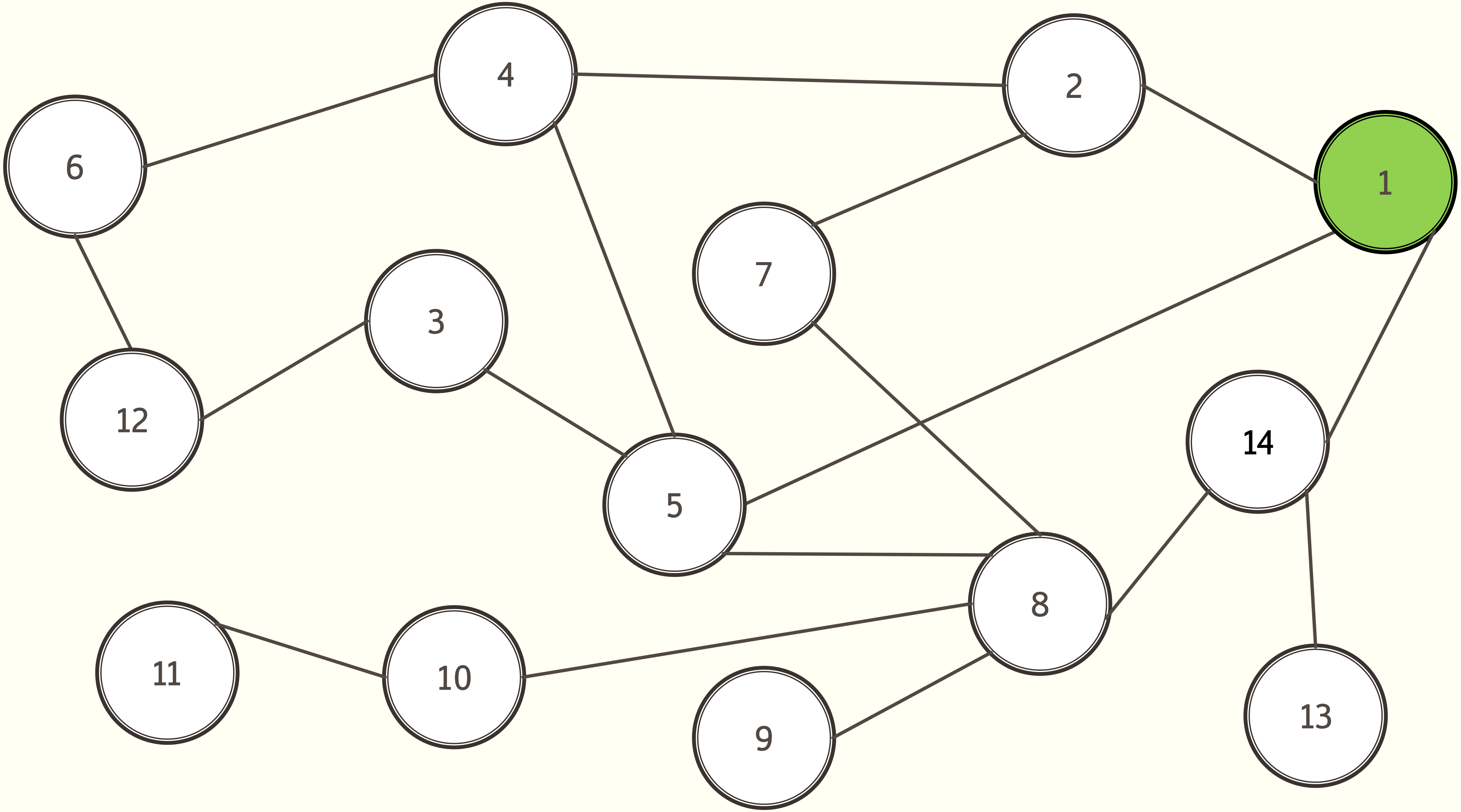
Gossip protocols

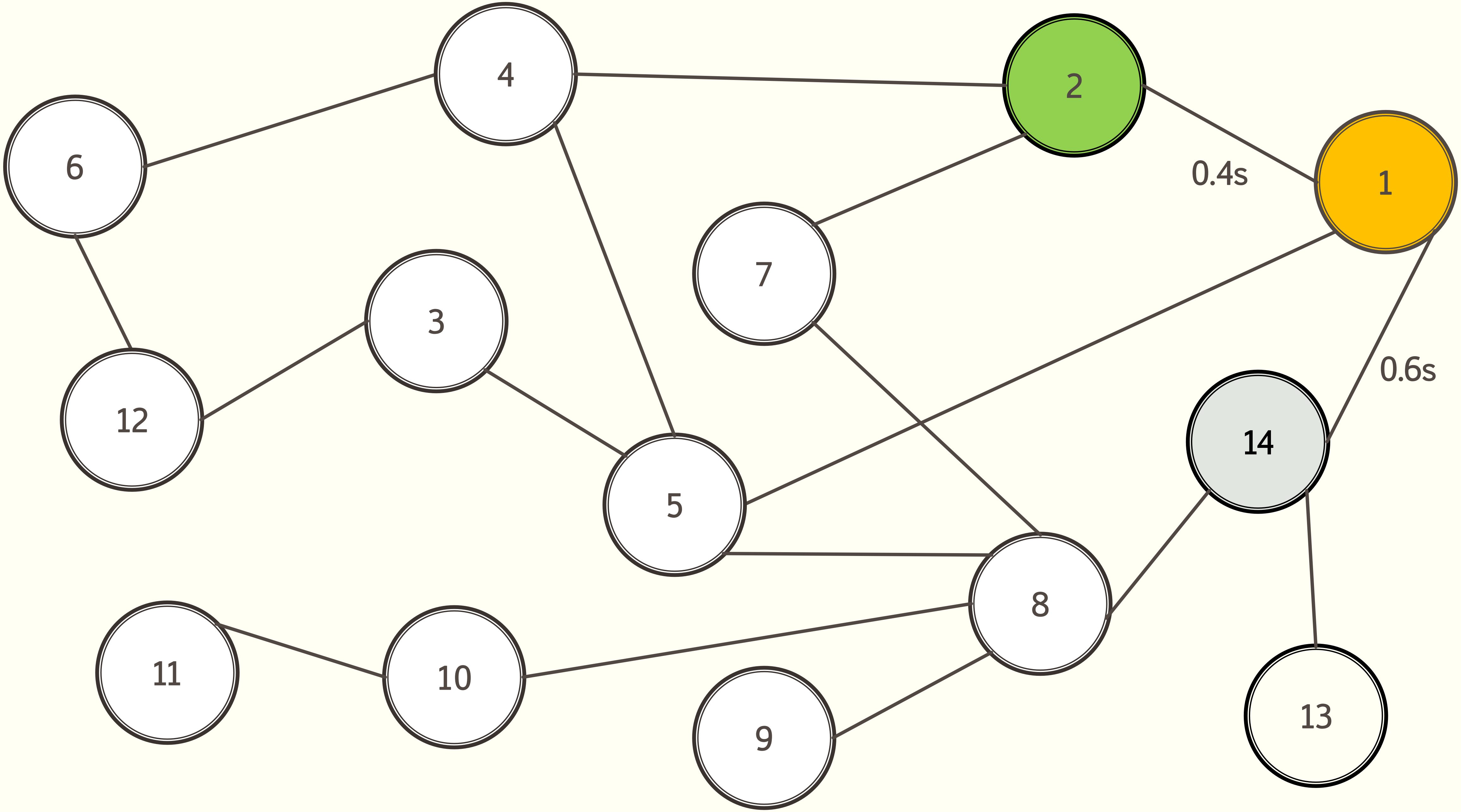
- **Reliability**: broadcast to entire network.
- A node send a small number of messages.
- **Fault-tolerant**
- Small number of rounds to reach the entire network.

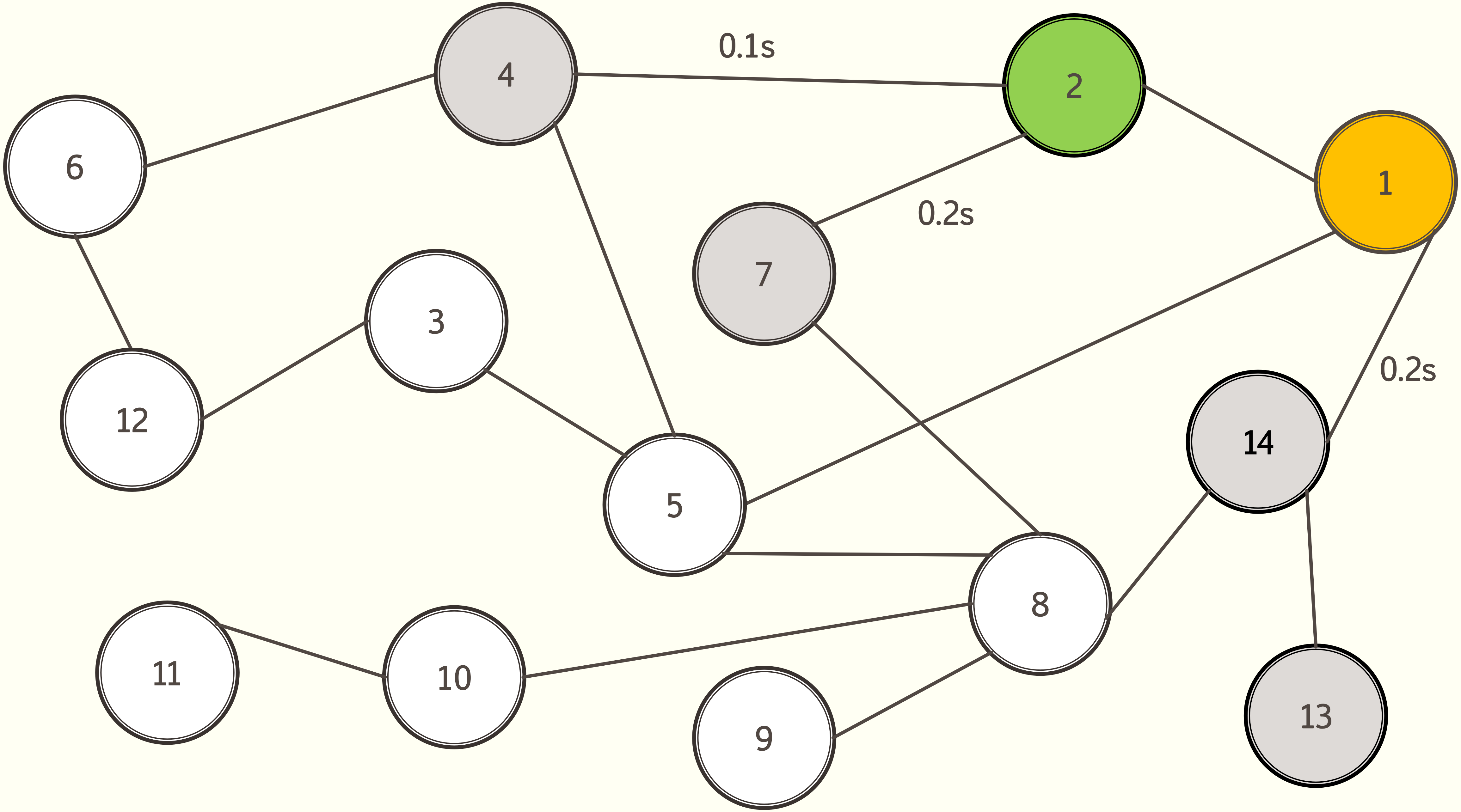
- A super node could keep track of every message.
- A super node may deanonymize Bitcoin transactions.

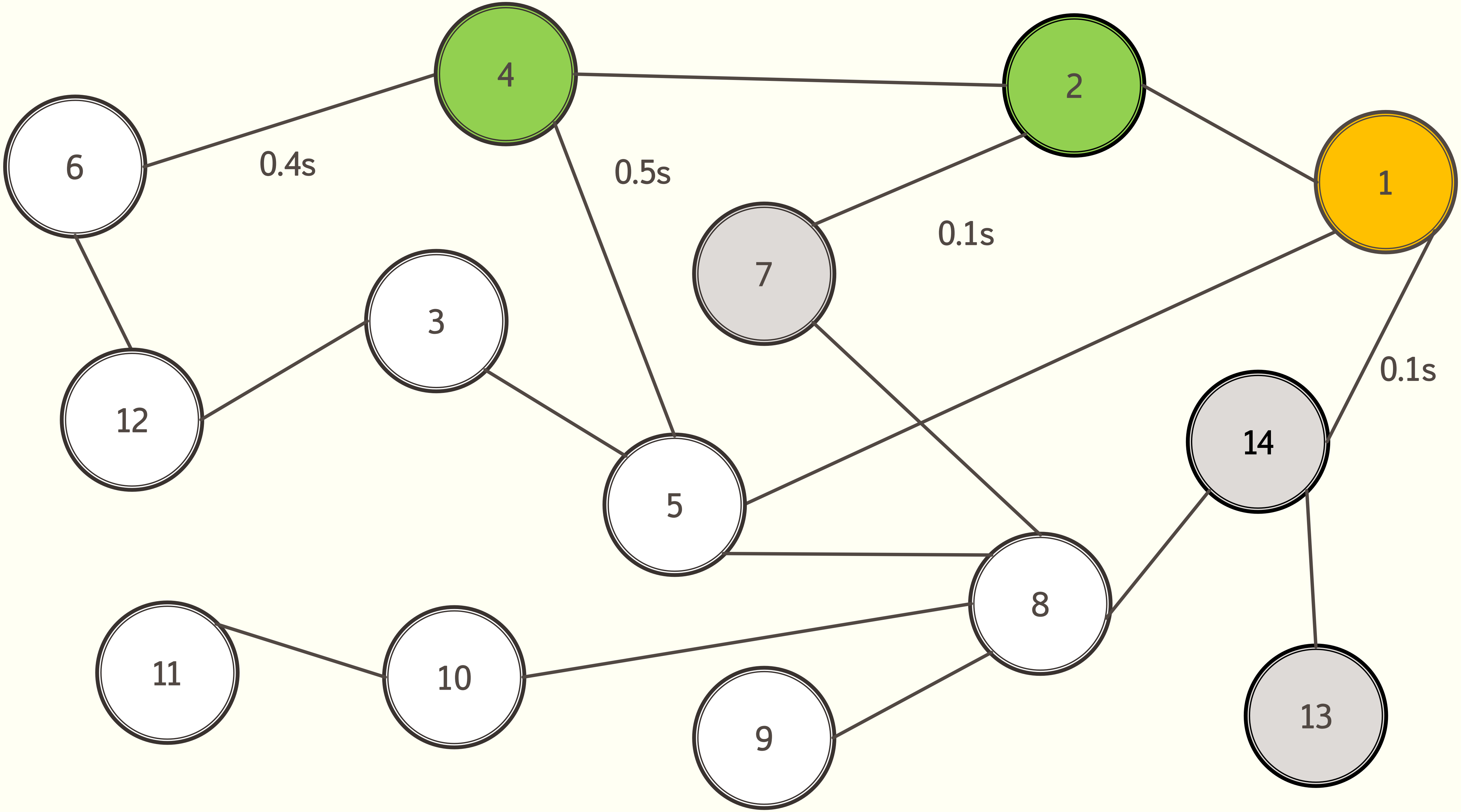
Gossip protocols -- diffusion

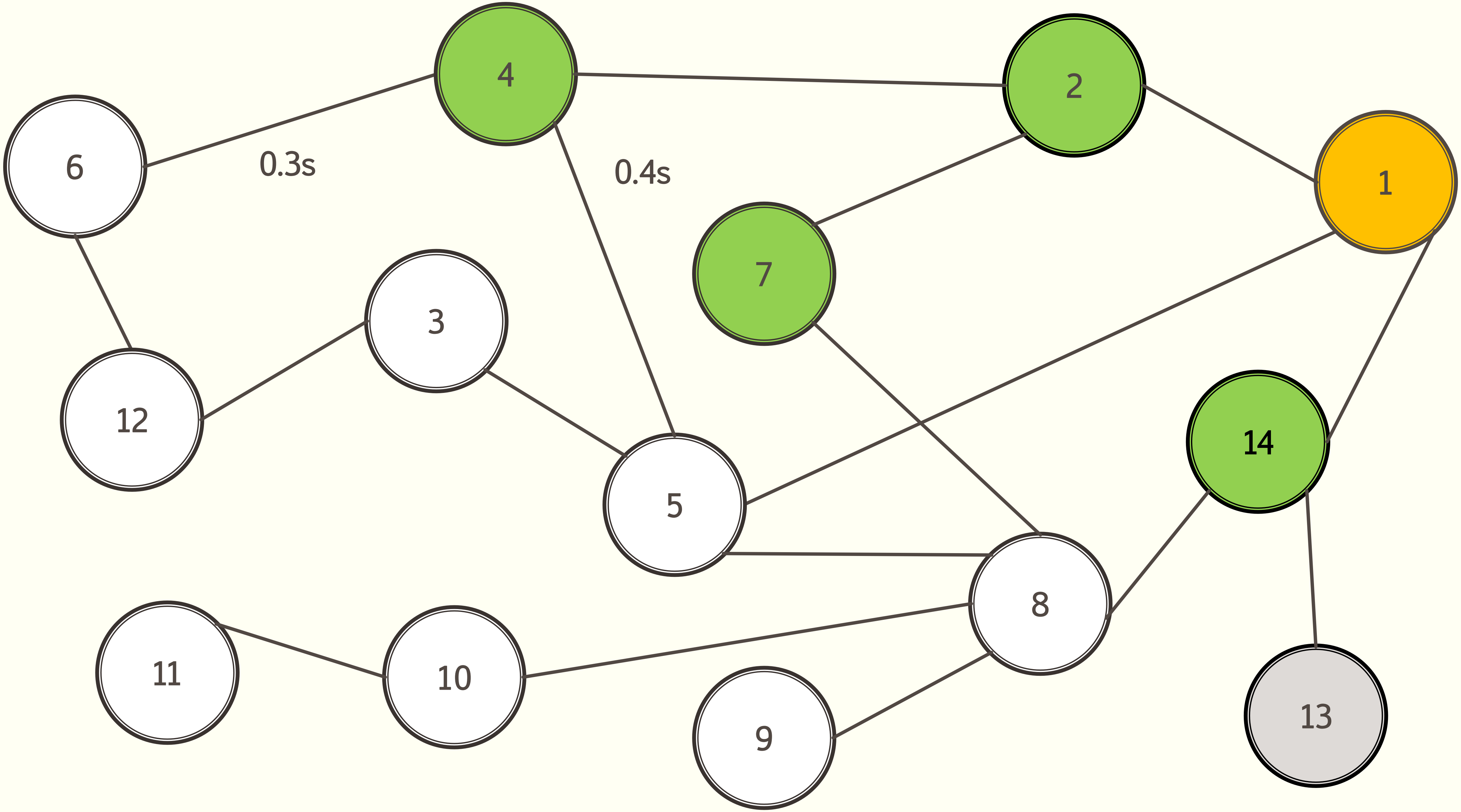
- Each node sends a message to K random targets (multicast).
- Each peer waits a random delay (exponential) before sending the message.


4
2
6
1
7
3
12
14
5
8
11
10
9
13


4
2
6
0.4s
1
7
3
0.6s
12
14
5
8
11
10
9
13


4
0.1s
2
6
1
0.2s
7
3
0.2s
12
14
5
8
11
10
9
13


4
2
1
6
0.4s
0.5s
7
0.1s
3
12
0.1s
14
5
8
11
10
9
13


4
2
6
0.3s
0.4s
1
7
3
12
14
5
8
11
10
9
13