Epidemic Algorithms for Replicated Database Maintenance

Shiang Chin
sc2983@cornell.edu

EPIDEMIC ALGORITHMS FOR REPLICATED DATABASE MAINTENANCE

Alan Demers, Dan Greene, Carl Hauser, Wes Irish, John Larson,
Scott Shenker, Howard Sturgis, Dan Swinehart, and Doug Terry

Epidemic Algorithms For Replicated Database Maintenance

- **Alan Demers** Retired Professor at Cornell University
- **Dan Greene** At Xerox PARC – Vehicle networks
- **Carl Hauser** Associate Professor, Washington State University
- **Wes Irish** Coyote Hill Consulting
- **Scott Shenker** Professor at UCBerkeley
- **Doug Terry** Microsoft Research

John Larson, Howard Sturgis, Dan Swinehart

Summary of the Research

- Database management for distributed systems
  - Consistent data records

- 3 methods
  - Direct Mail
  - Anti-Entropy
  - Rumor Mongering

- CAP Theorem

- Real world applications
  - Vegvisir blockchain
  - Amazon

Research Motivation

- Clearinghouse servers on Xerox Corporate Internet (CIN)
    - Hundreds of ethernets connected by gateways and phone lines
        - Ex Message: Japan -> Europe goes through 14 gateways and 7 phone lines
    - Organized by Hierarchical name (domains)
    - Remailing – Inefficiency during disagreement among participants

Points of Differentiation

- Eventual delivery of repeated messages and do not require data structures at one server to describe information held at other servers
- Algorithm are randomized

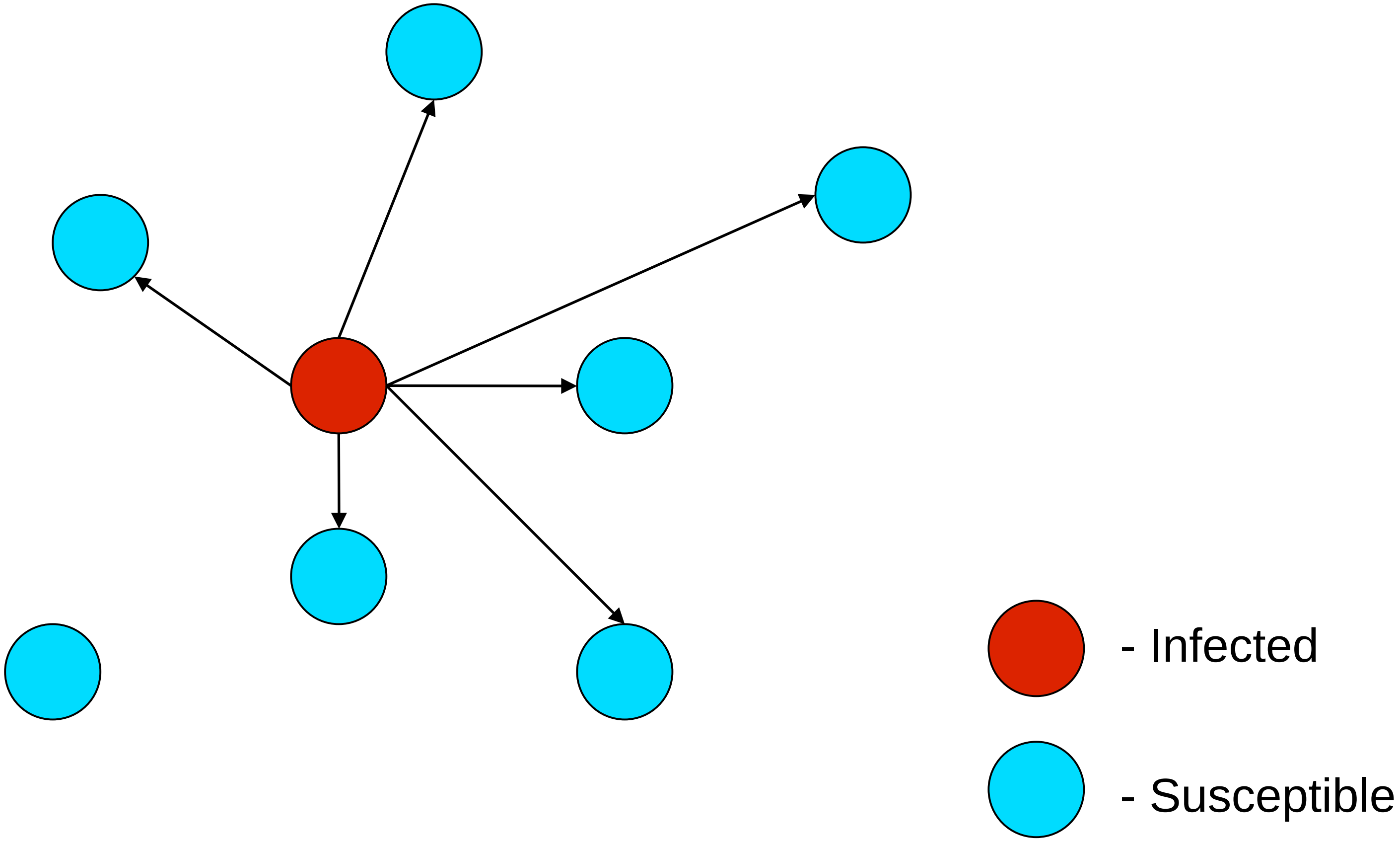
Vocabulary

- **Infected** – Knows the update and *spreads* it
- **Susceptible** – Does not know the update
- **Removed** – Knows the update but *not* able to spread it anymore
- **Push** – Tells an updates to another node
- **Pull** – Asks for an update from another node

Direct Mail

Direct Mail – Sends update to all nodes in the network

- Traffic proportional to the number of sites * average distance between sites



Direct Mail

Failure Modes

- Message discarded for nodes
  - Que overflows
  - Extended period of inaccessibility

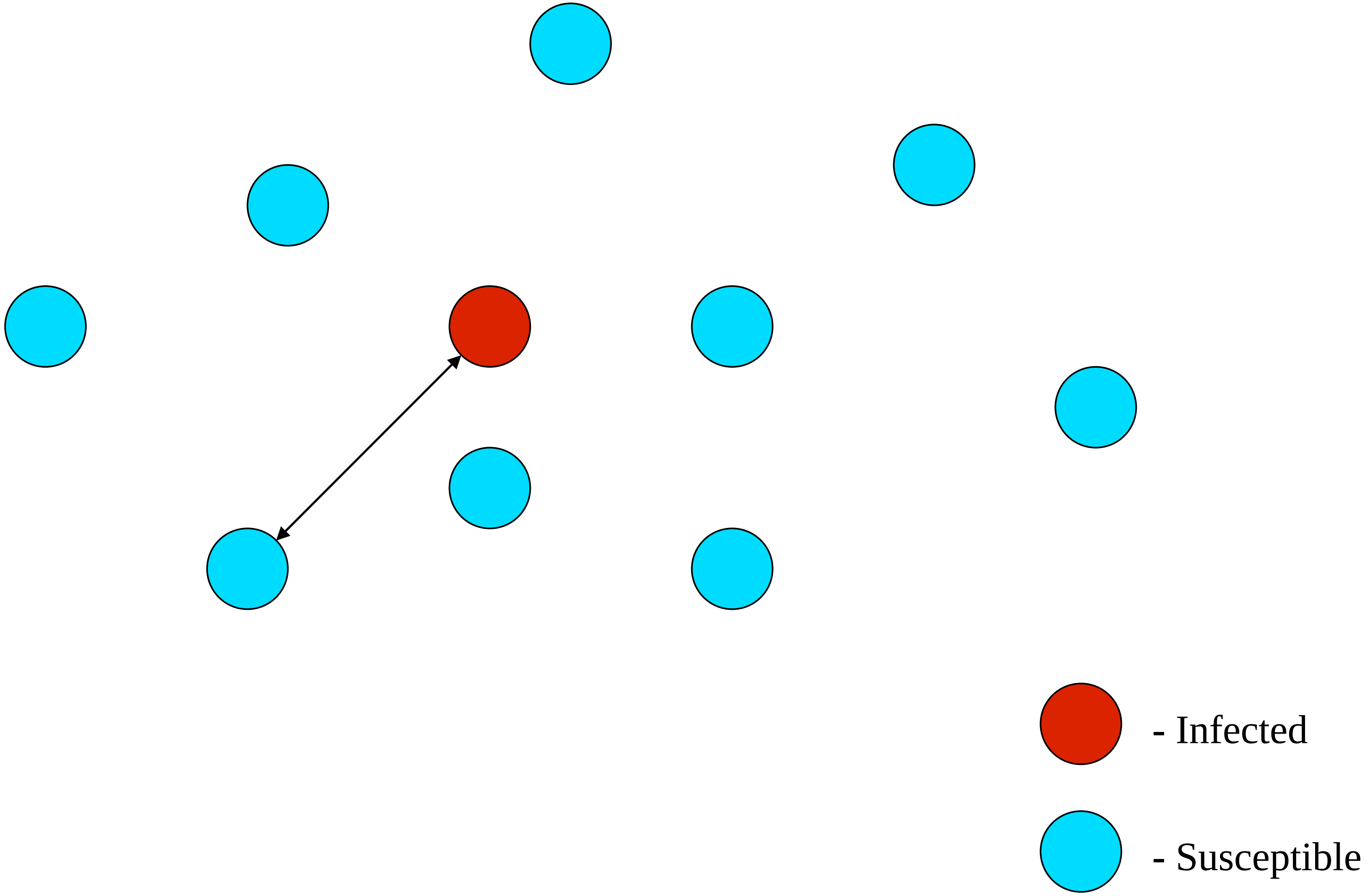
Anti-Entropy

Anti-Entropy – Nodes exchange messages with a random node through the methods below:

- Pull – Grows fast but slows down overtime
- Push – Grows slowly but speeds up overtime
- Push-pull – Most efficient and every node receives the message

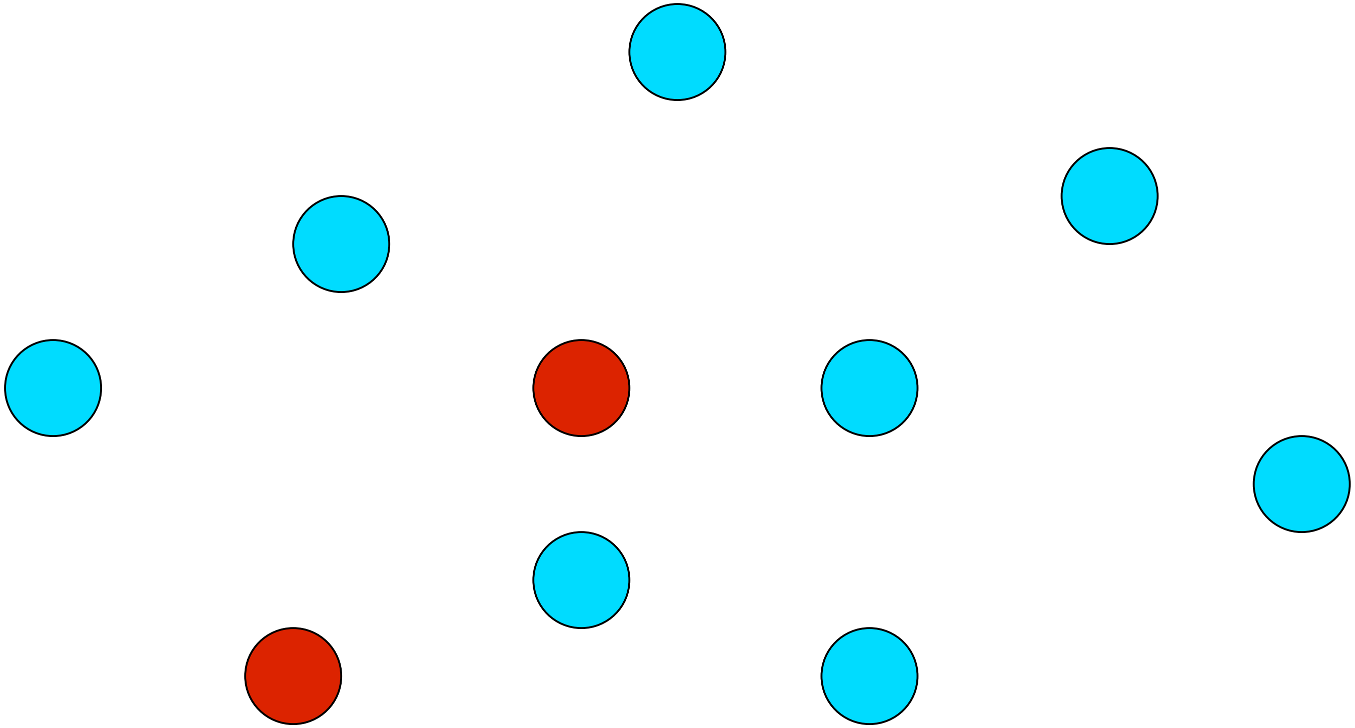
How it works

Anti-Entropy



How it works

Anti-Entropy

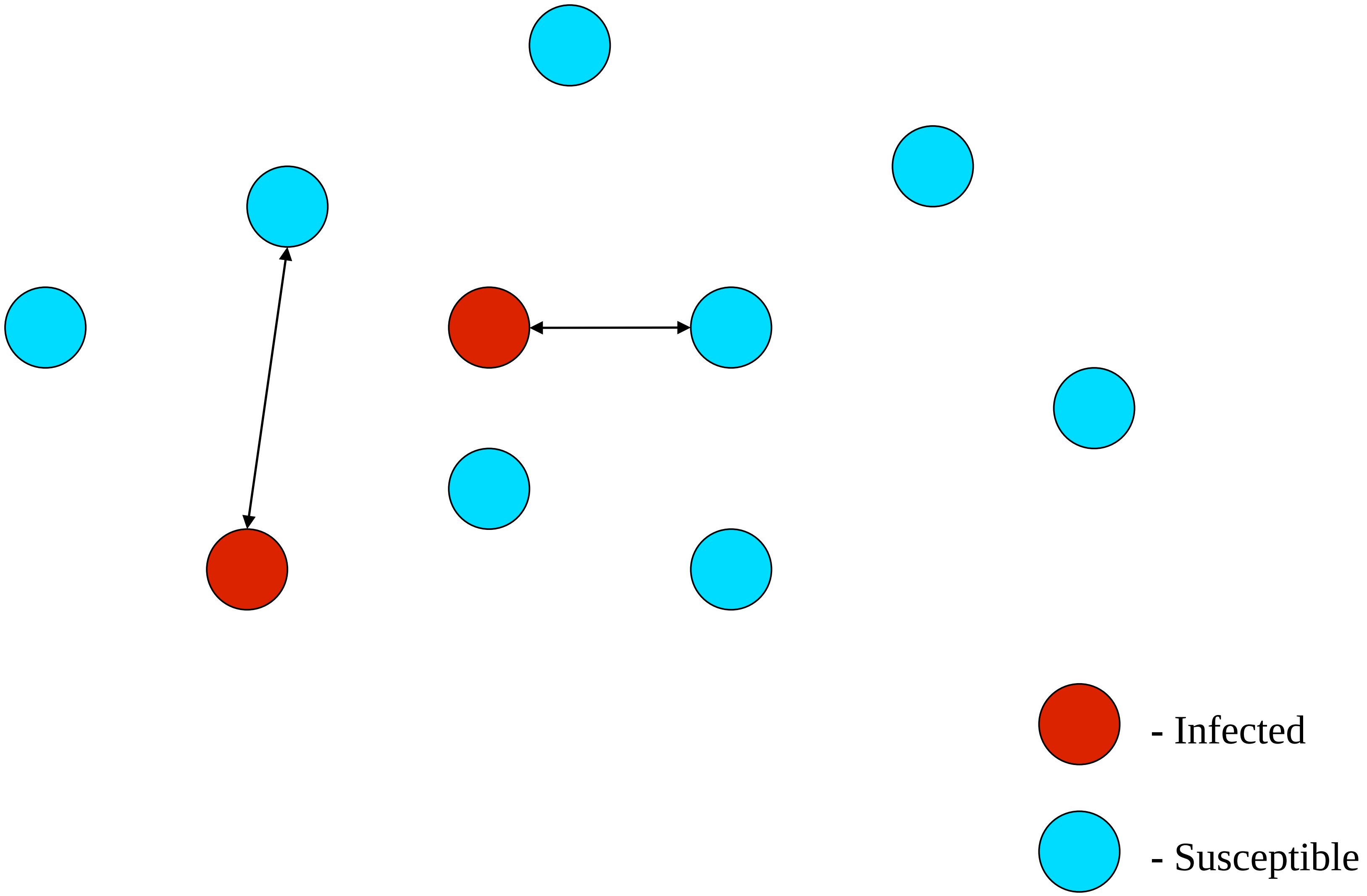


🔴 - Infected

🔵 - Susceptible

How it works

Anti-Entropy

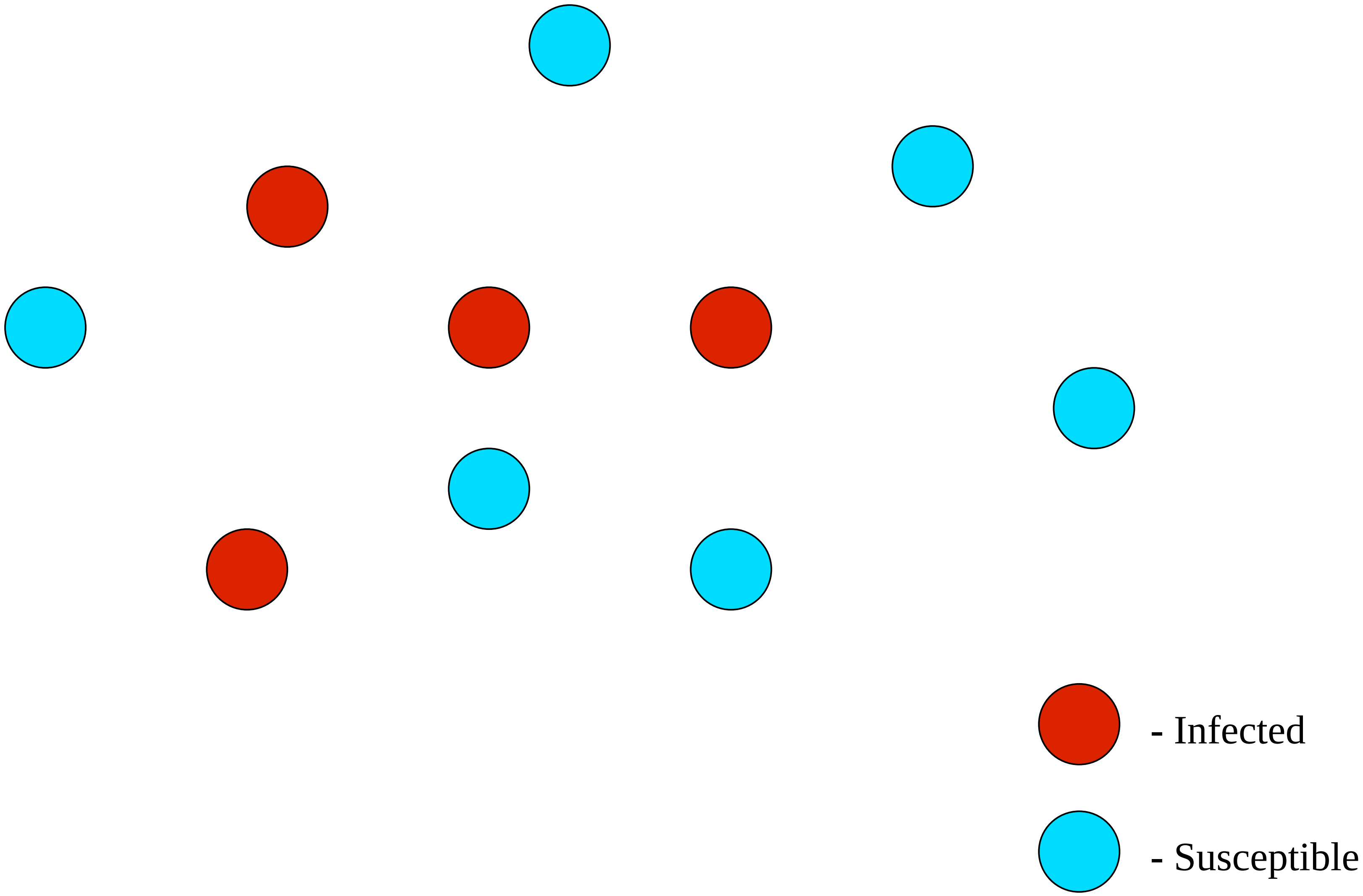


🔴 - Infected

🔵 - Susceptible

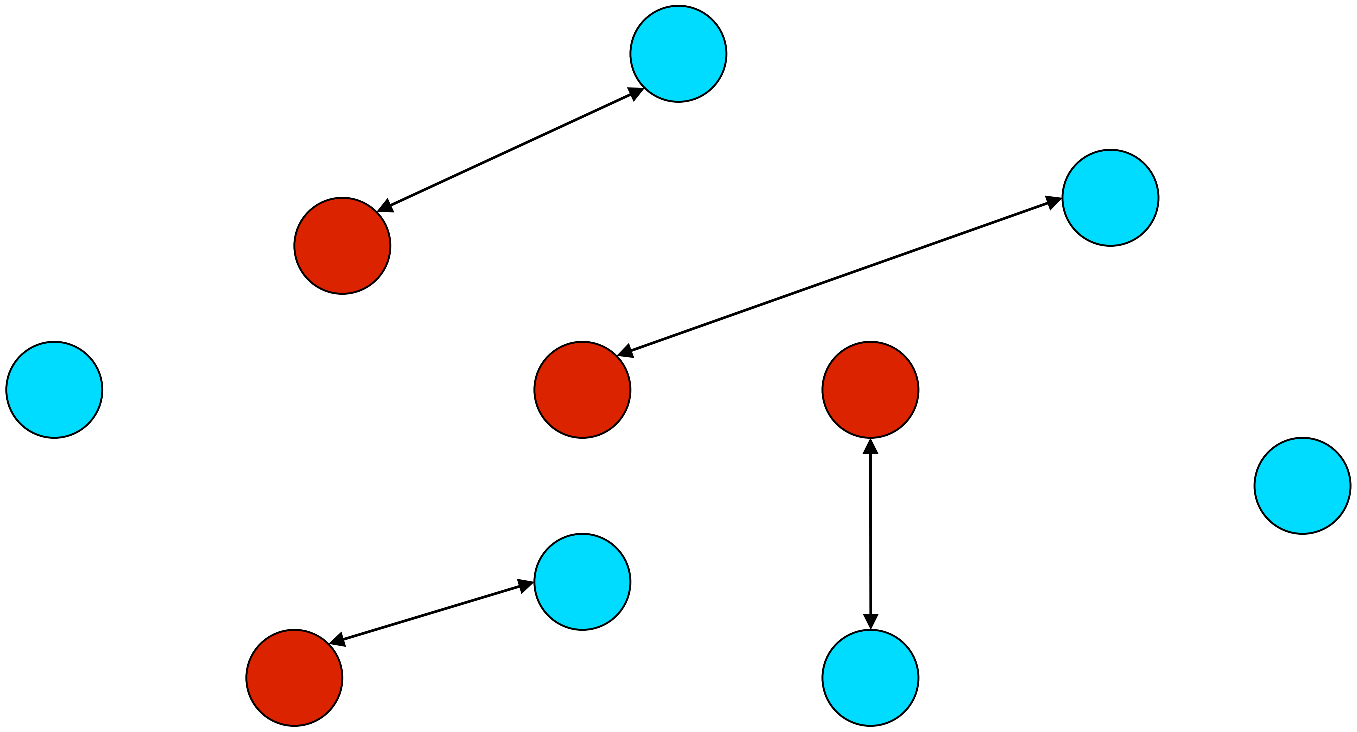
How it works

Anti-Entropy



How it works

Anti-Entropy

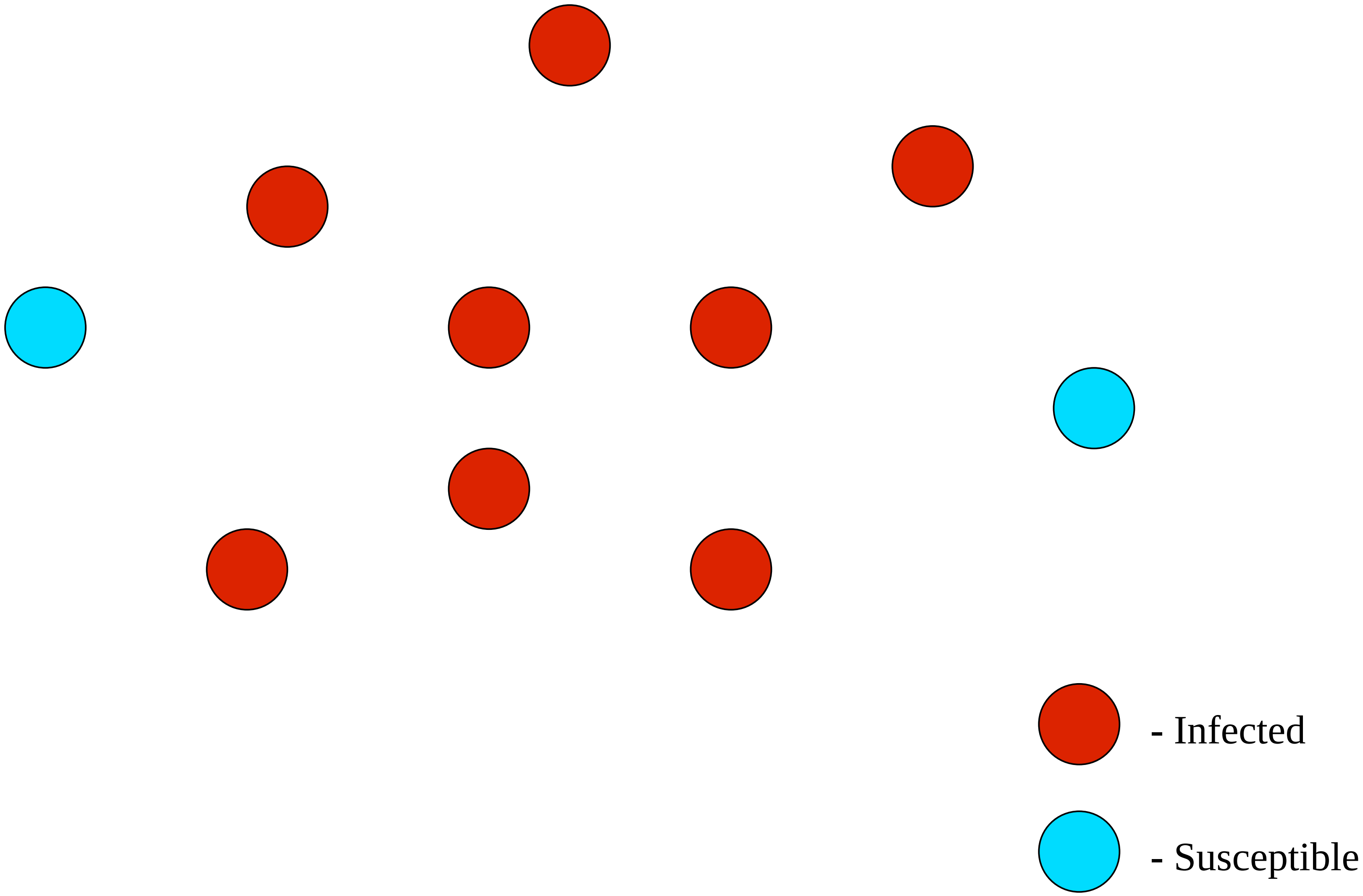


🔴 - Infected

🔵 - Susceptible

How it works

Anti-Entropy



Anti-Entropy

Pro:

- Eventually everyone receives the message

Con:

- Large overhead due to external requests for updates

Rumor Mongering

Rumor mongering – Optimized algorithm for spreading messages

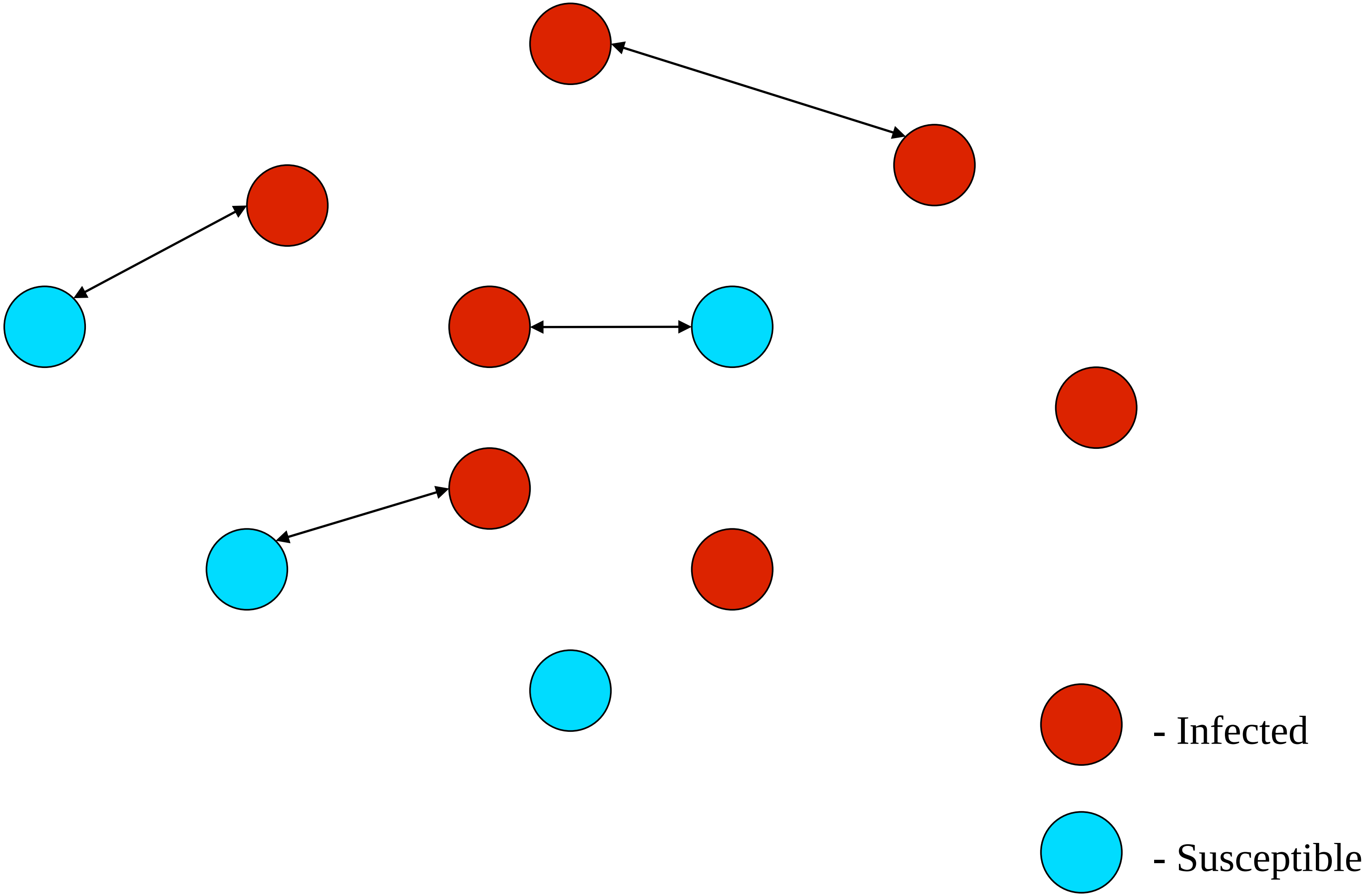
- When a node receives a new update (rumor)
- Periodically choose another site at random to infect other nodes
- When enough nodes have seen the rumor it is removed

Problem of convergence

- Fix with anti-entropy combination

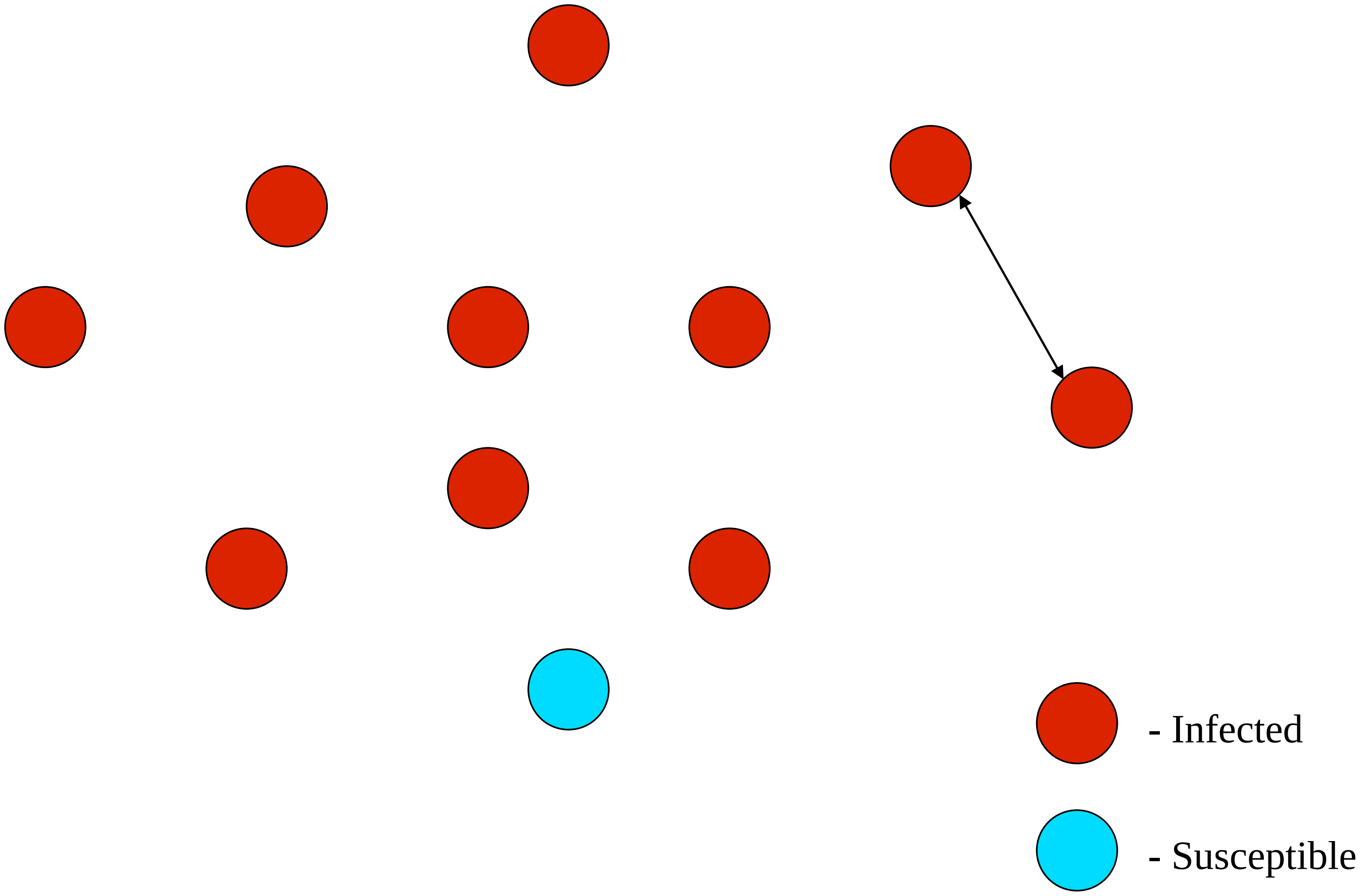
How it works

Rumor mongering



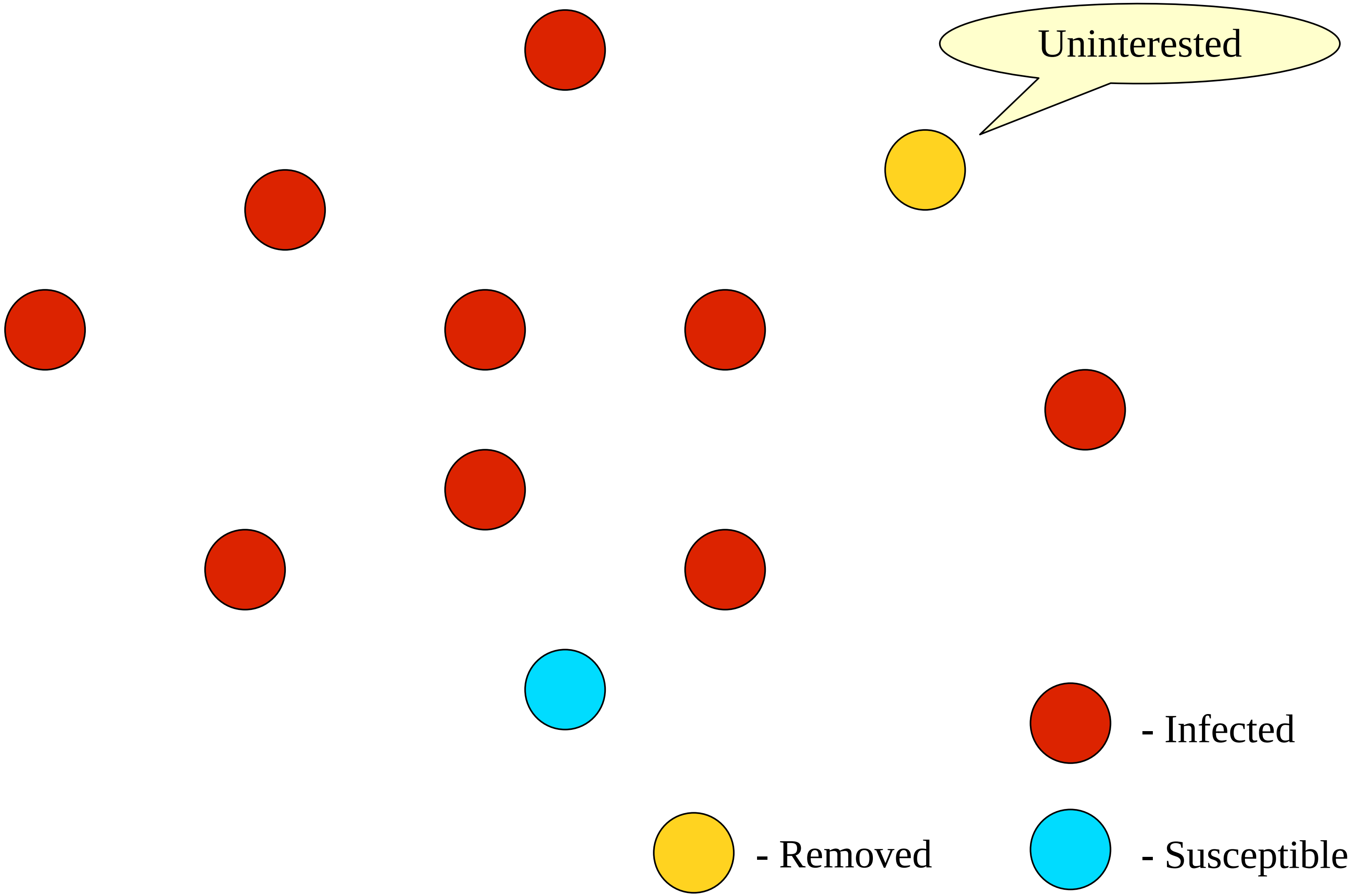
How it works

Rumor mongering



How it works

Rumor mongering



Uninterested

🔴 - Infected

🟡 - Removed

🔵 - Susceptible

Points of differentiation

Death certificates

- Shows when a node is decommissioned
- Verified with a timestamp

Spatial distribution

- Favors sending updates to closest nodes first

Anti-Entropy Results

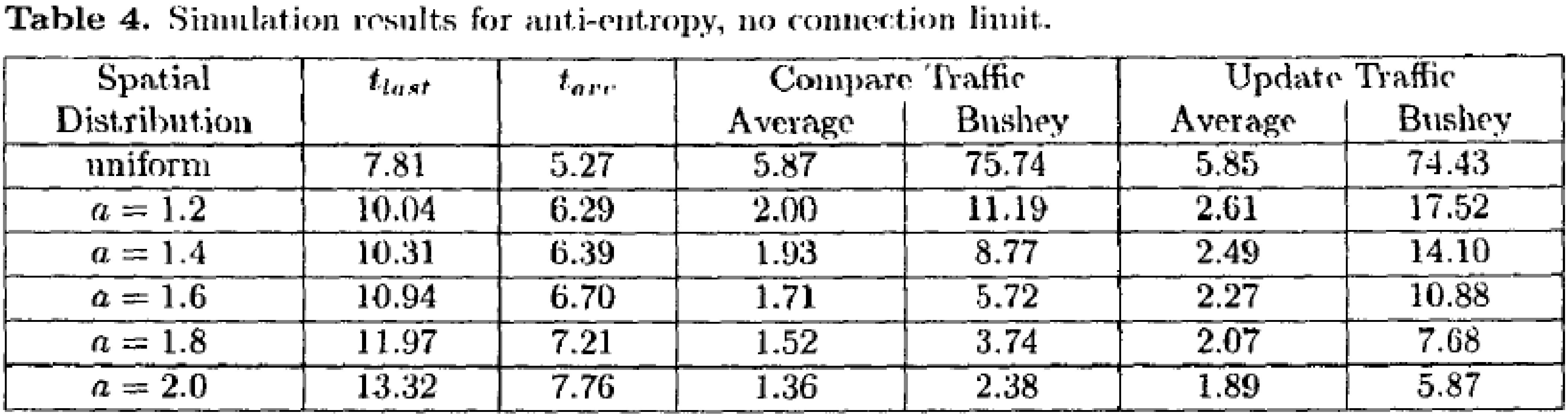


Table 4. Simulation results for anti-entropy, no connection limit.

Spatial Distribution	t_{last}	t_{ave}	Compare Traffic		Update Traffic	
			Average	Bushey	Average	Bushey
uniform	7.81	5.27	5.87	75.74	5.85	74.43
$a = 1.2$	10.04	6.29	2.00	11.19	2.61	17.52
$a = 1.4$	10.31	6.39	1.93	8.77	2.49	14.10
$a = 1.6$	10.94	6.70	1.71	5.72	2.27	10.88
$a = 1.8$	11.97	7.21	1.52	3.74	2.07	7.68
$a = 2.0$	13.32	7.76	1.36	2.38	1.89	5.87

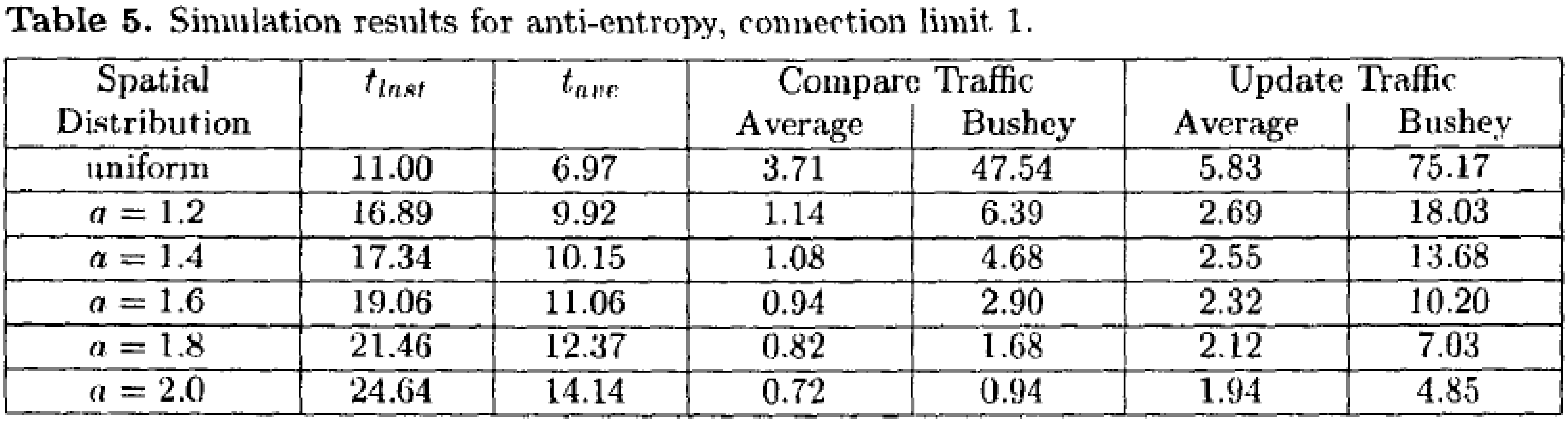


Table 5. Simulation results for anti-entropy, connection limit 1.

Spatial Distribution	t_{last}	t_{ave}	Compare Traffic		Update Traffic	
			Average	Bushey	Average	Bushey
uniform	11.00	6.97	3.71	47.54	5.83	75.17
$a = 1.2$	16.89	9.92	1.14	6.39	2.69	18.03
$a = 1.4$	17.34	10.15	1.08	4.68	2.55	13.68
$a = 1.6$	19.06	11.06	0.94	2.90	2.32	10.20
$a = 1.8$	21.46	12.37	0.82	1.68	2.12	7.03
$a = 2.0$	24.64	14.14	0.72	0.94	1.94	4.85

Rumor Mongering Results

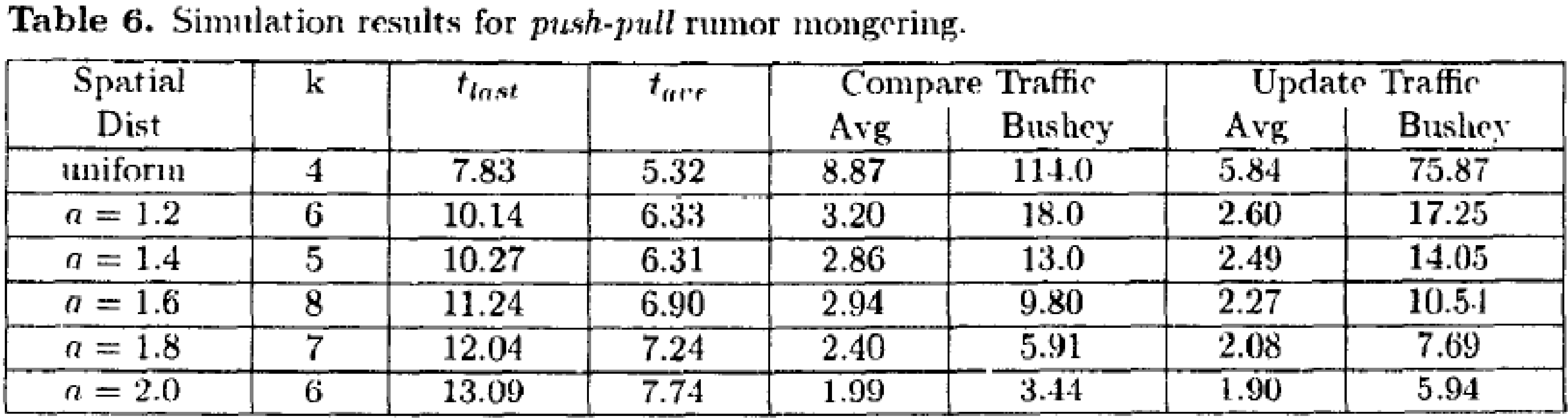


Table 6. Simulation results for *push-pull* rumor mongering.

Spatial Dist	k	t_{last}	t_{ave}	Compare Traffic		Update Traffic	
				Avg	Bushey	Avg	Bushey
uniform	4	7.83	5.32	8.87	114.0	5.84	75.87
$\alpha = 1.2$	6	10.14	6.33	3.20	18.0	2.60	17.25
$\alpha = 1.4$	5	10.27	6.31	2.86	13.0	2.49	14.05
$\alpha = 1.6$	8	11.24	6.90	2.94	9.80	2.27	10.54
$\alpha = 1.8$	7	12.04	7.24	2.40	5.91	2.08	7.69
$\alpha = 2.0$	6	13.09	7.74	1.99	3.44	1.90	5.94

CAP Theorem

Only 2/3 are achievable

- Consistency – Every node has the most recent message
- Accessibility – Every node receives a message but no guarantee that it is the most recent
- Partition Tolerance – System continues to operate even if messages are lost

Applications of the Research

1. Vegvisir – Agriculture specific blockchain that reconciles with random nodes within a specific range
2. Amazon – S3 storage system uses gossip to disseminate information

Questions?