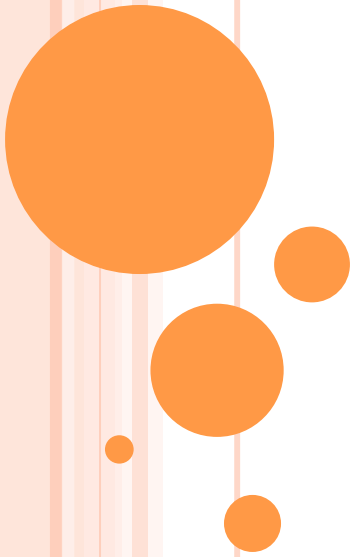


EPIDEMIC TECHNIQUES

Ki Suh Lee

OUTLINE

- Epidemic Protocol
- Epidemic Algorithms for Replicated Database Maintenance
- Astrolabe: A Robust and scalable technology for distributed system monitoring, management, and data mining

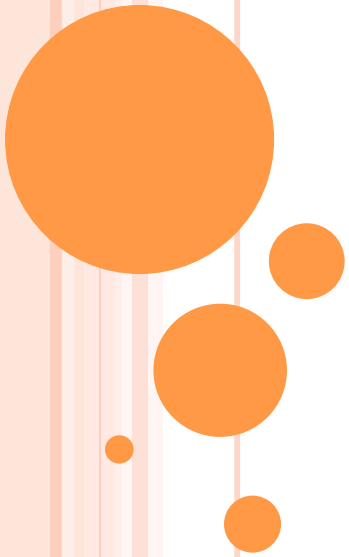
EPIDEMIC PROTOCOL

- *Gossip Protocol*: efficient and robust for data dissemination
- From Mathematical Theory of Epidemiology
- Direct-mail: Each update is mailed to all sites
- Anti-entropy: Every site regularly chooses another site at random and resolves differences
- Rumor mongering: Periodically chooses another site at random and ensures the update. When a site has tried to share a hot rumor with **too many sites** that have already seen it, *stop propagating it*

EXAMPLES OF GOSSIP

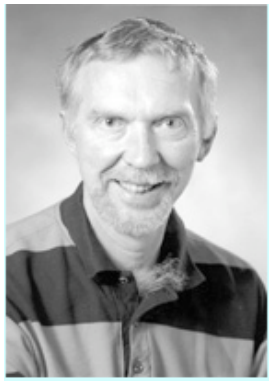
- *Anti-entropy*
  - Bayou(Xerox, 1995)
  - Failure Detection Systems(Cornell, 1998)
  - Bimodal Multicast(Cornell, 1998)
  - Astrolabe(Cornell, 1999)
  - Lightweight Probabilistic Broadcast(2003)
  - Kelips(Cornell, 2003)
  - Bamboo(Berkeley, 2003)
- *Rumor Mongering*
  - Trickle(2004)

EPIDEMIC ALGORITHM FOR REPLICATED DATABASE MAINTENANCE



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

AUTHORS



Alan Demers
Cornell Univ.

Dan Greene
Palo Alto
Research Center

Carl Hauser
Washington State
Univ.

Wesley Irish
Coyote Hill
Consulting LLC

Scott Shenker
EECS Berkley

Doug Terry
MS Research
Silicon Valley

John Larson
Howard Sturgis
Dan Swinehart

MOTIVATION

- Clearinghouse Servers on Xerox Corporate Internet(CIN) (1987)
  - Several thousand workstations, servers, and computing hosts connected by gateways and phone lines
- Performance problems due to traffic created to achieve consistency on replicated domains
- Used direct mail and anti-entropy.
- Anti-entropy couldn't be completed because of the load on the network.

INTRODUCTION

- Goal:
  - To design algorithms for distributing updates and driving replicated database toward consistency
  - Efficient, robust, and scalable

- Two Important Considerations
  - The time for an update to propagate to all sites
  - The network traffic generated in propagating a single update

EPIDEMIC ALGORITHMS

- Direct Mail
- Anti-entropy
- Rumor mongering

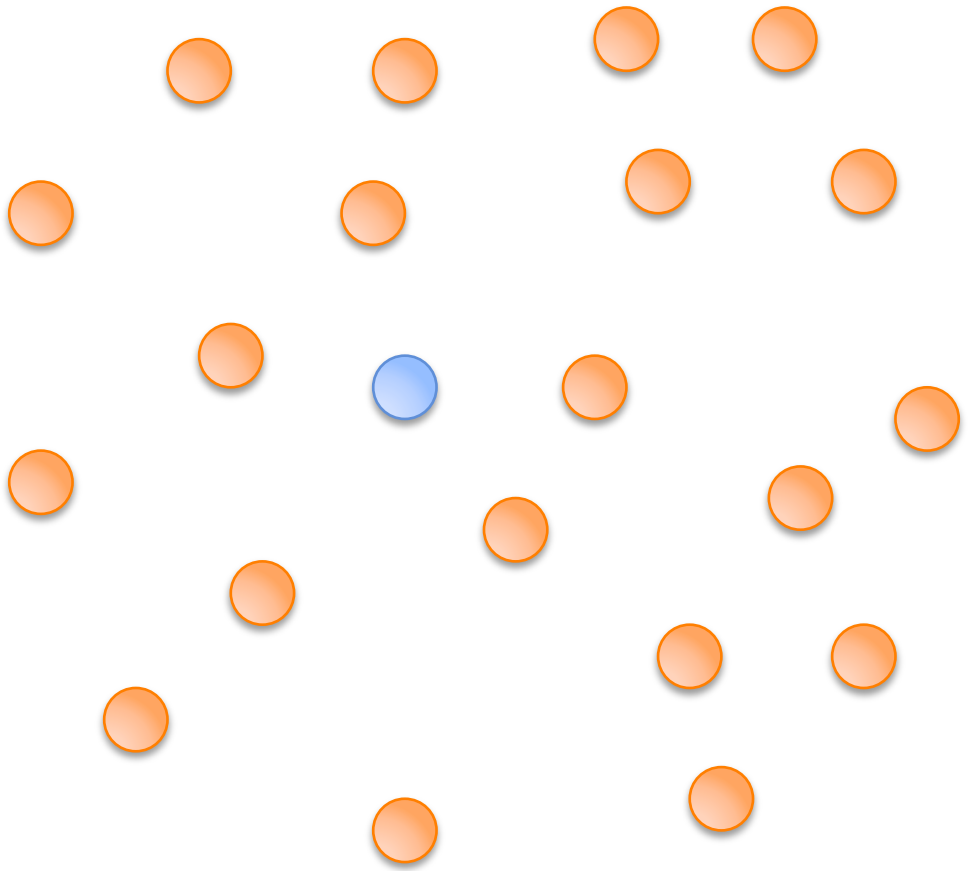
TERMINOLOGY

- "infective": a site holding an update it is willing to share
- "susceptible": a site that has not yet received an update
- "removed": a site that has received an update but is no longer willing to share

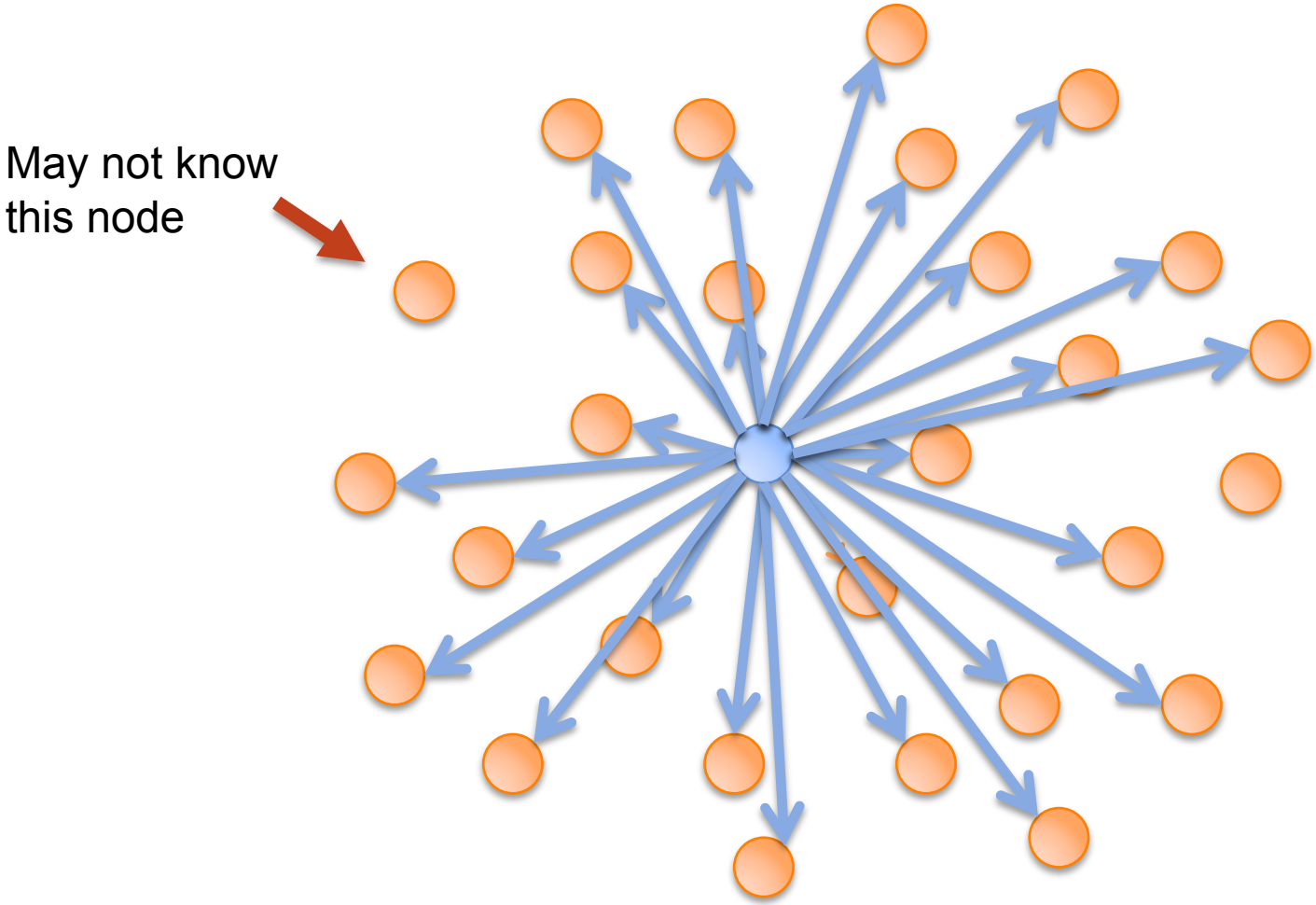
DIRECT MAIL

- Notify all other sites of an update
- When update received, check timestamp
  - Timely and reasonably efficient
- Not completely reliable.
  - Messages may be discarded
  - Source may not know all other nodes
- n messages per update

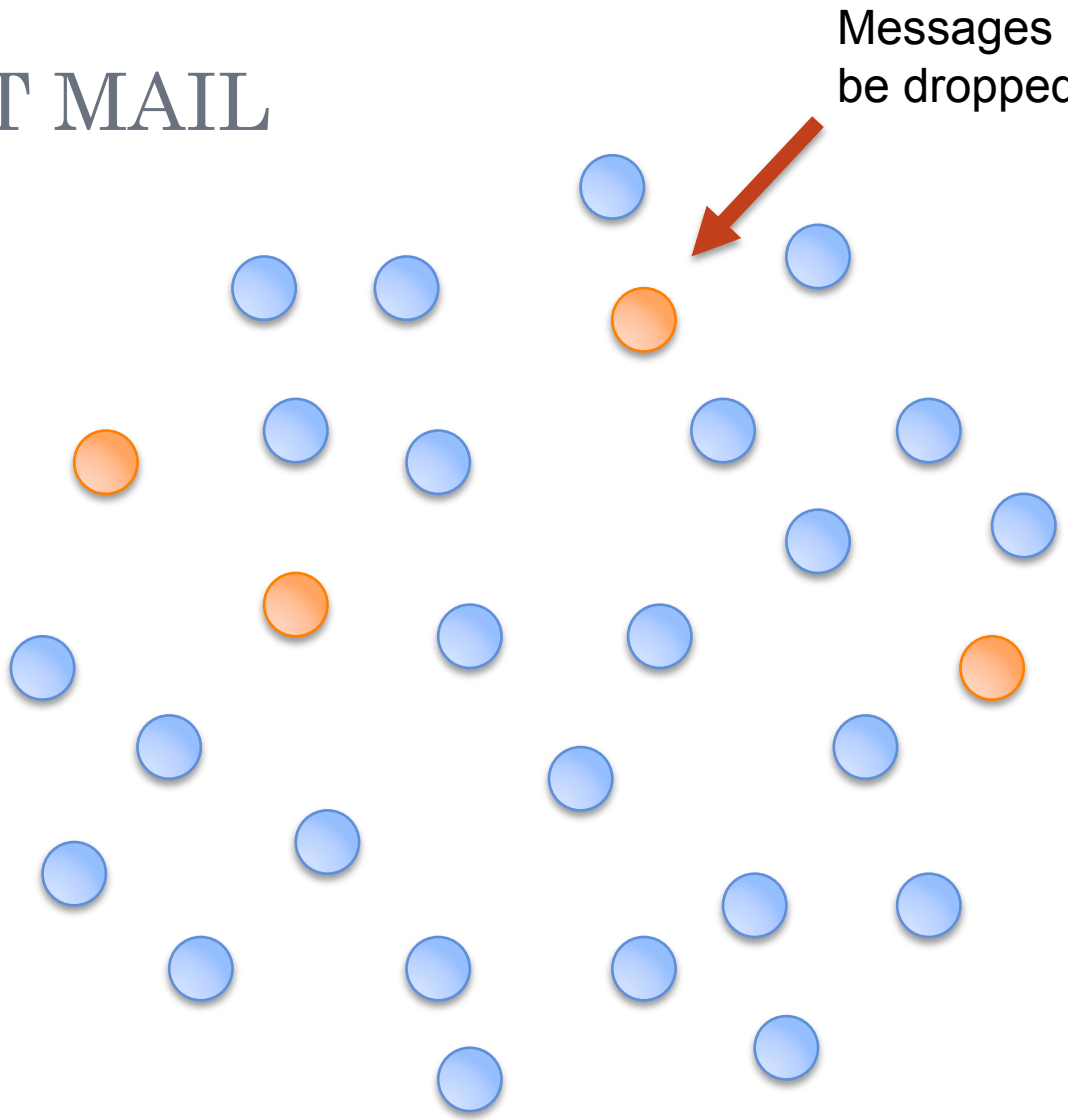
DIRECT MAIL



DIRECT MAIL



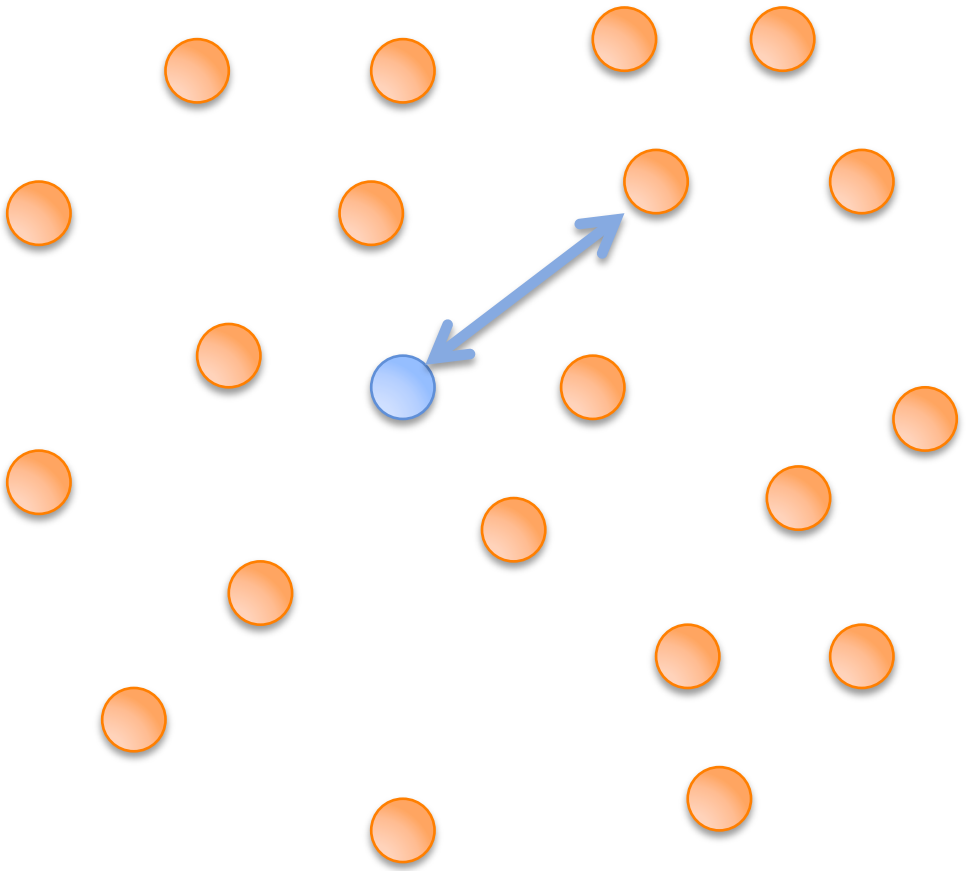
DIRECT MAIL



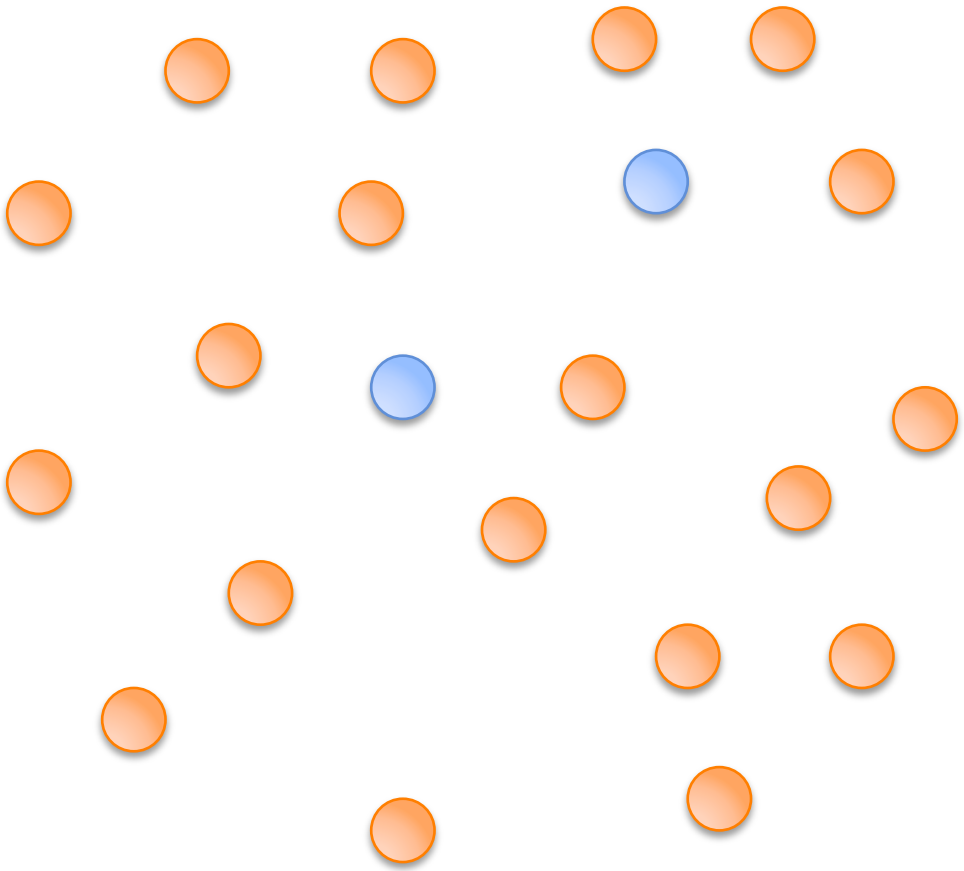
ANTI-ENTROPY(SIMPLE EPIDEMIC)

- every site regularly chooses another site at random
- by exchanging content, resolves any differences between them
  - Eventually infect the entire population
  - Extremely reliable, but expensive
  - Propagates more slowly than direct mail
  - O(log n) to eventually infect all nodes
- Differences are resolved using:
  - Push: infective -> susceptible
  - Pull: susceptible -> infective
  - Push-pull: depending on timestamp, chooses either push or pull

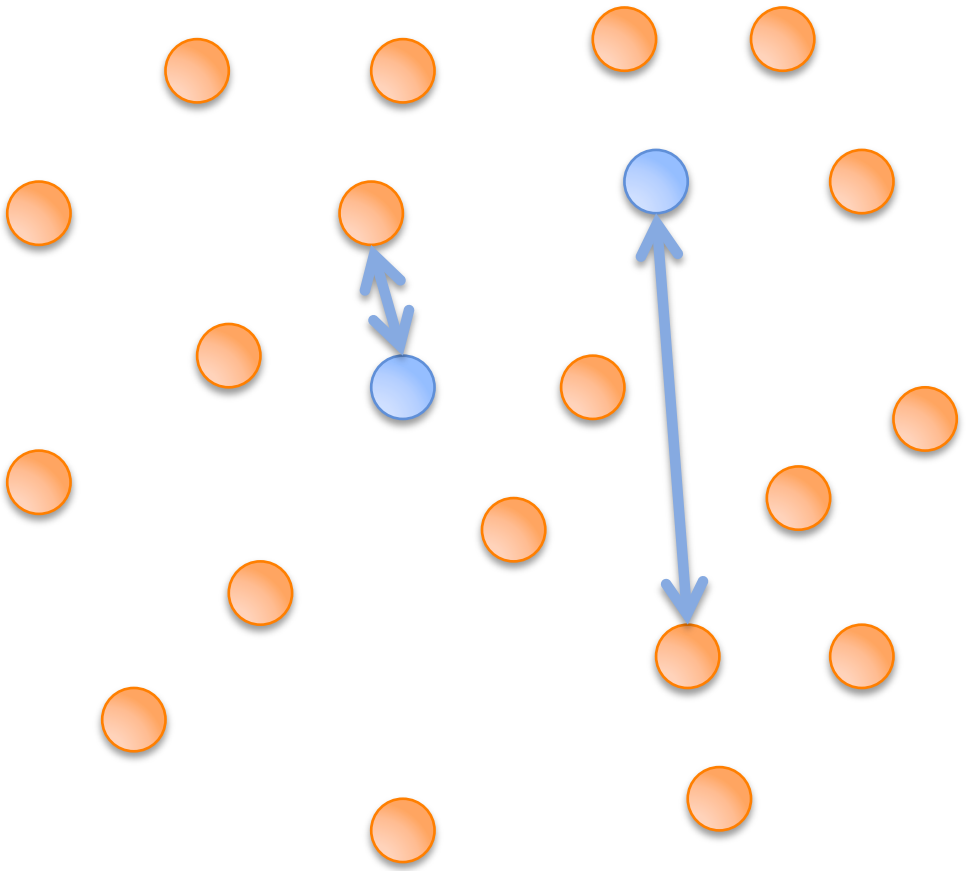
ANTI-ENTROPY



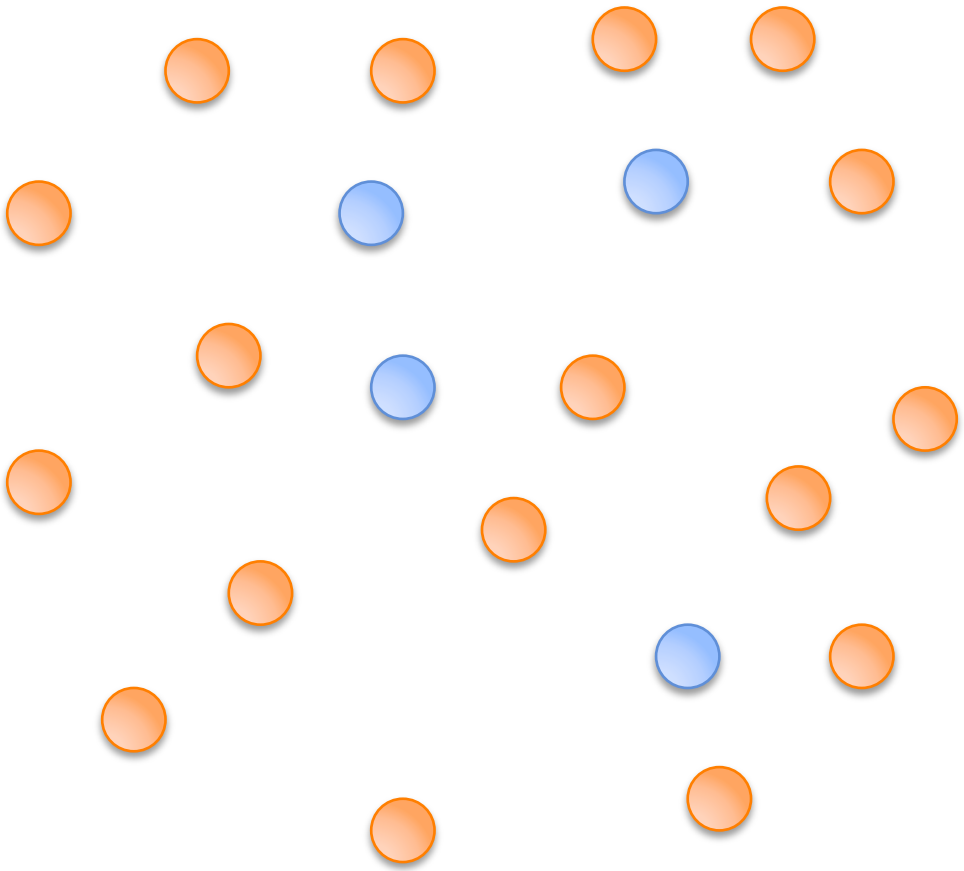
ANTI-ENTROPY



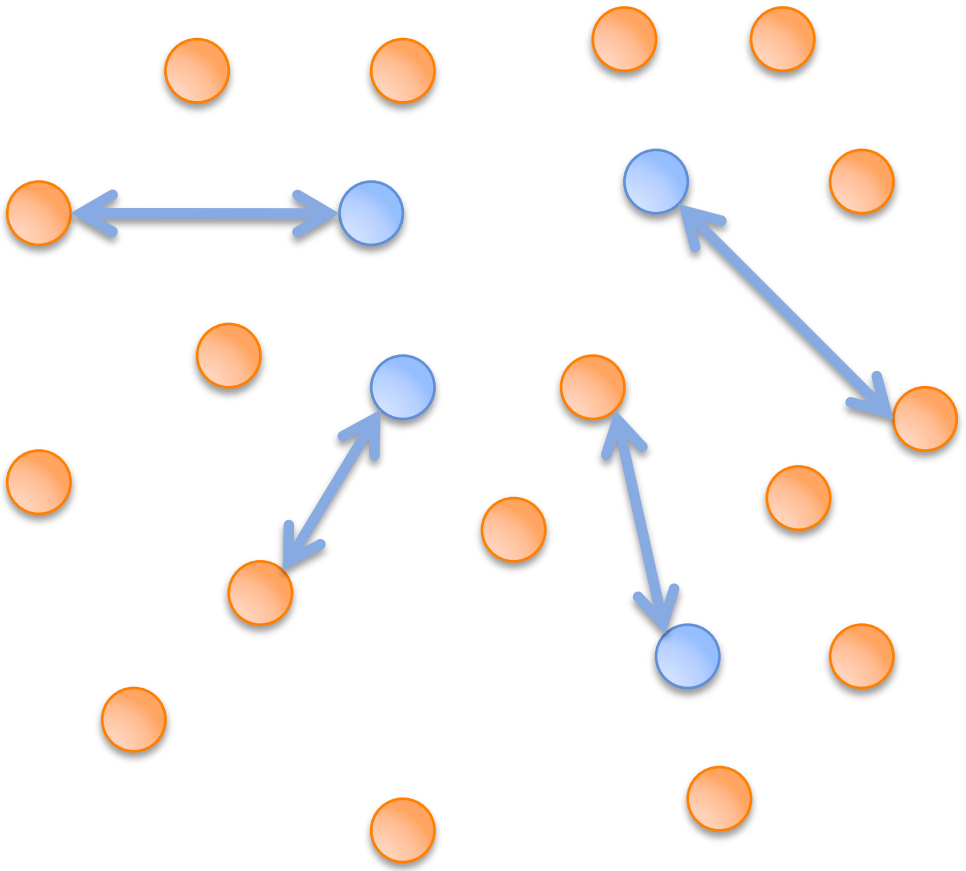
ANTI-ENTROPY



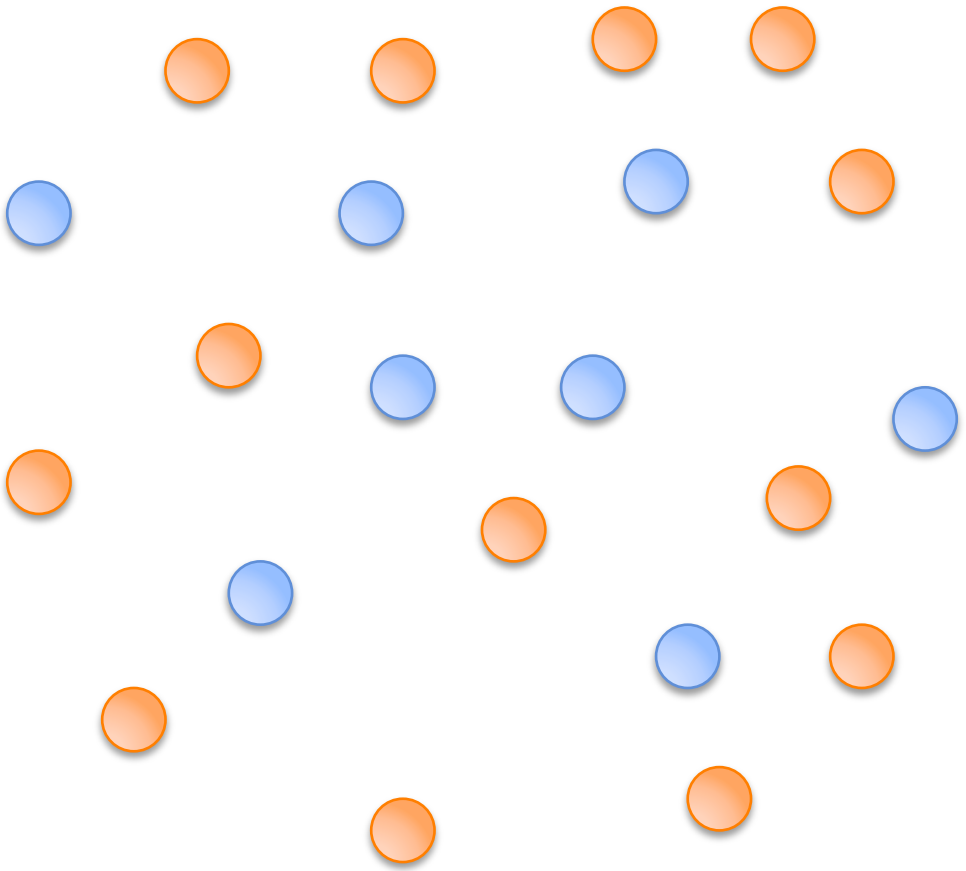
ANTI-ENTROPY



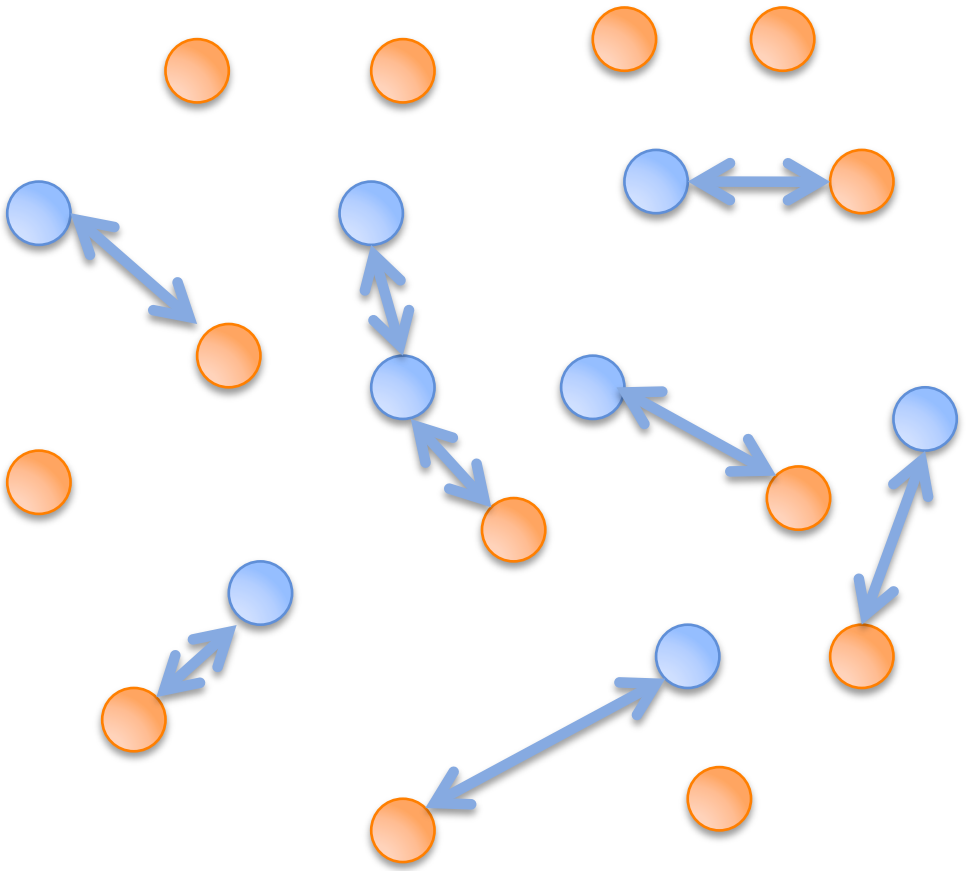
ANTI-ENTROPY



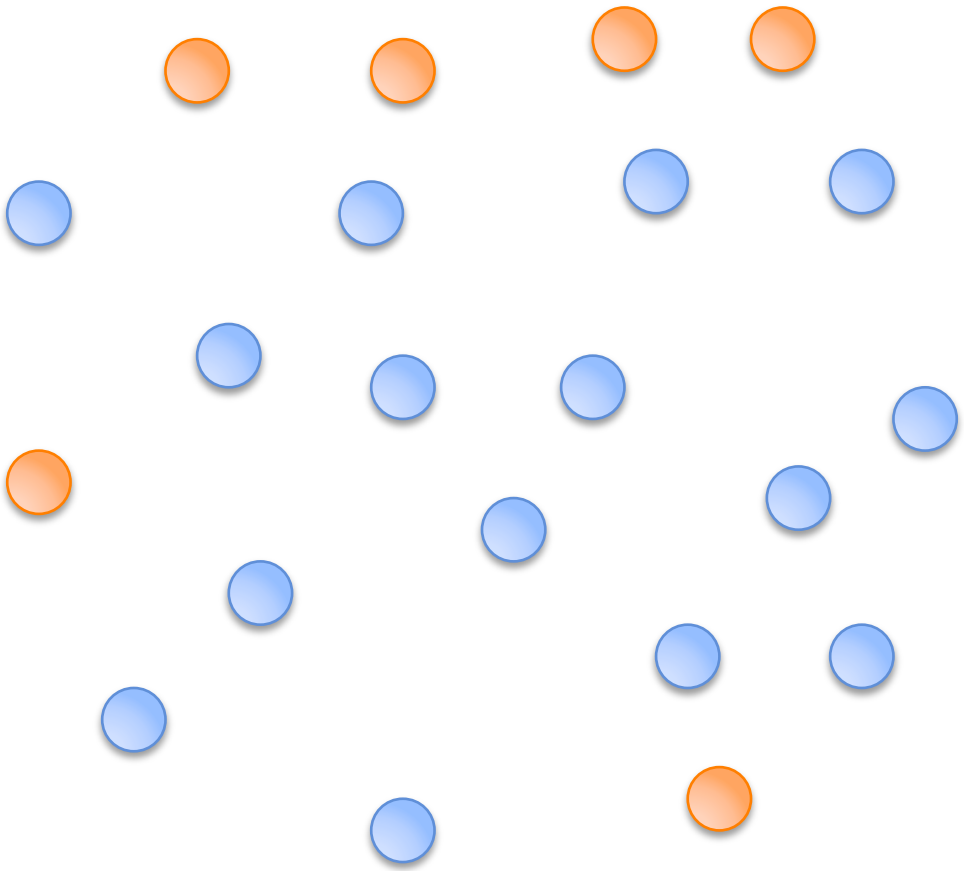
ANTI-ENTROPY



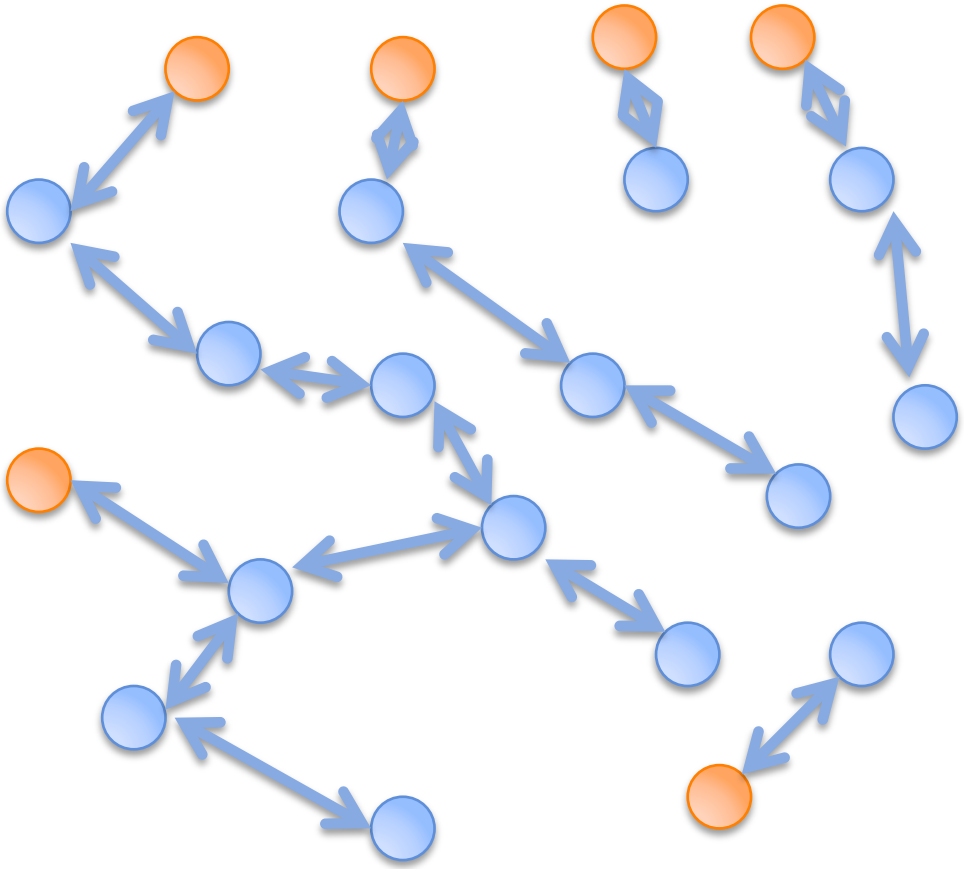
ANTI-ENTROPY



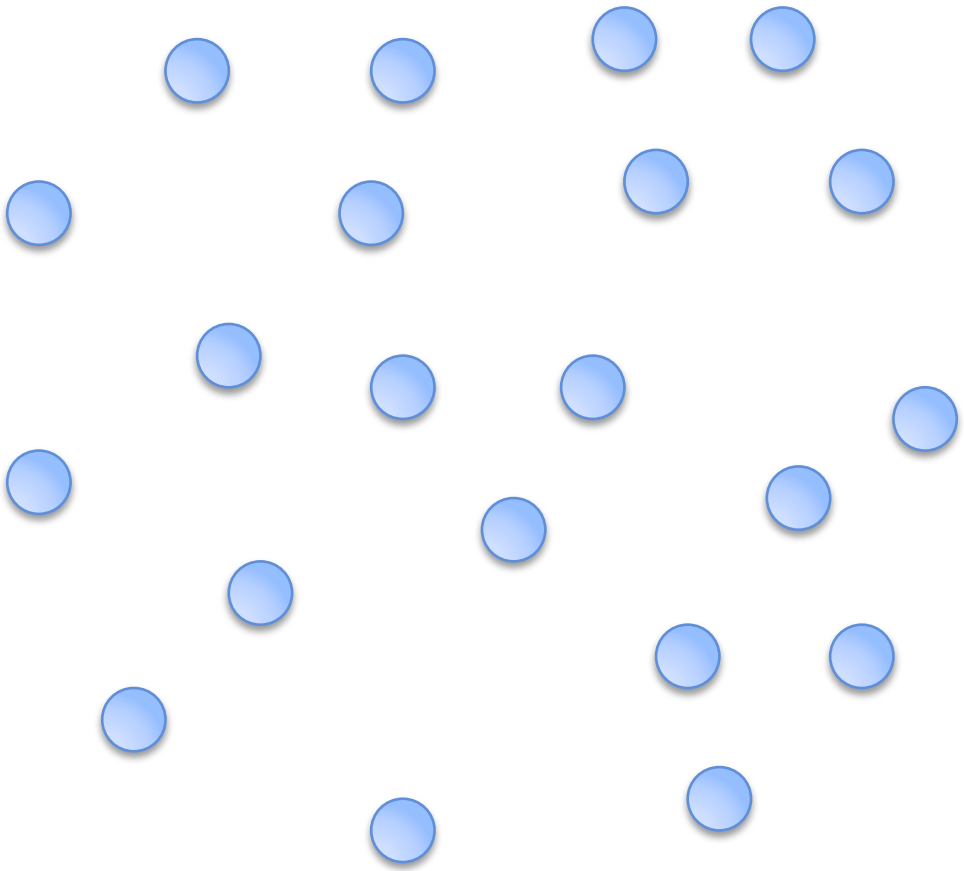
ANTI-ENTROPY



ANTI-ENTROPY



ANTI-ENTROPY



PUSH VS. PULL

- p_i – Probability that a node is susceptible after the i^{th} round
- Pull: $p_{i+1} = p_i^2$
- Push: $p_{i+1} = p_i\left(1-\frac{1}{n}\right)^{n(1-p_i)} = p_i\, e^{-1}$
- Pull converges faster than push

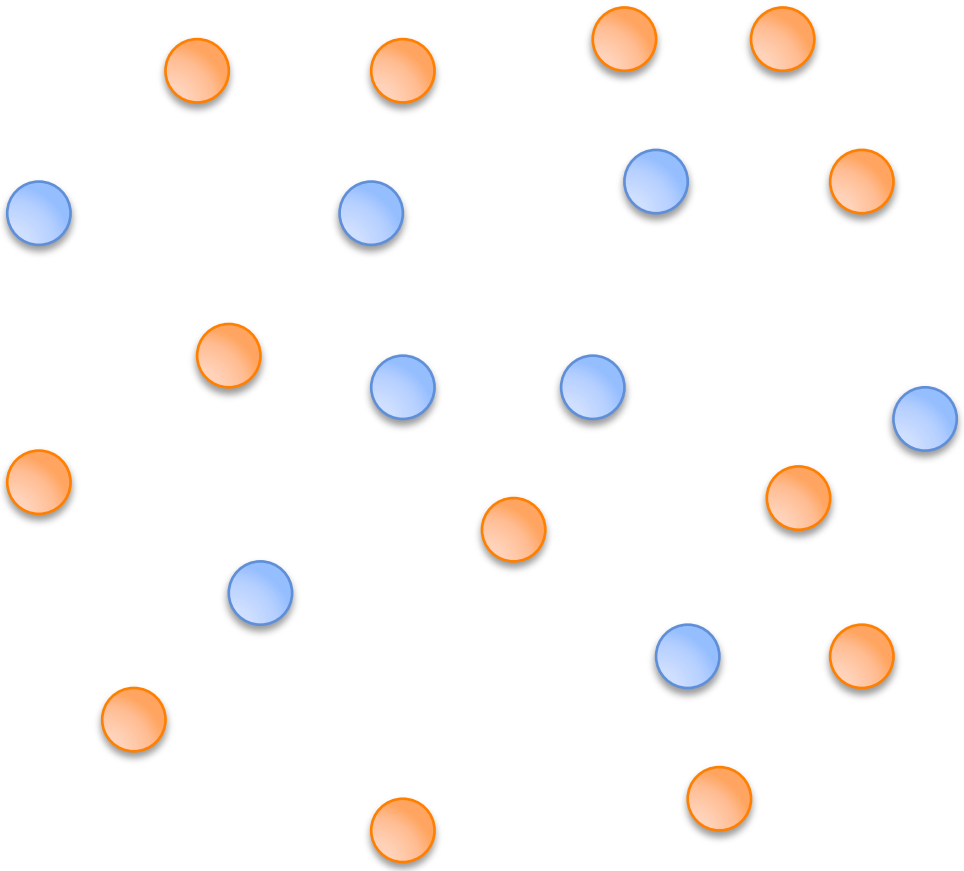
ANTI-ENTROPY: OPTIMIZATIONS

- Comparing two complete copies is expensive
- For performance improvement, we can use
  - Checksum
  - Recent Update Lists
  - Inverted Index by timestamp

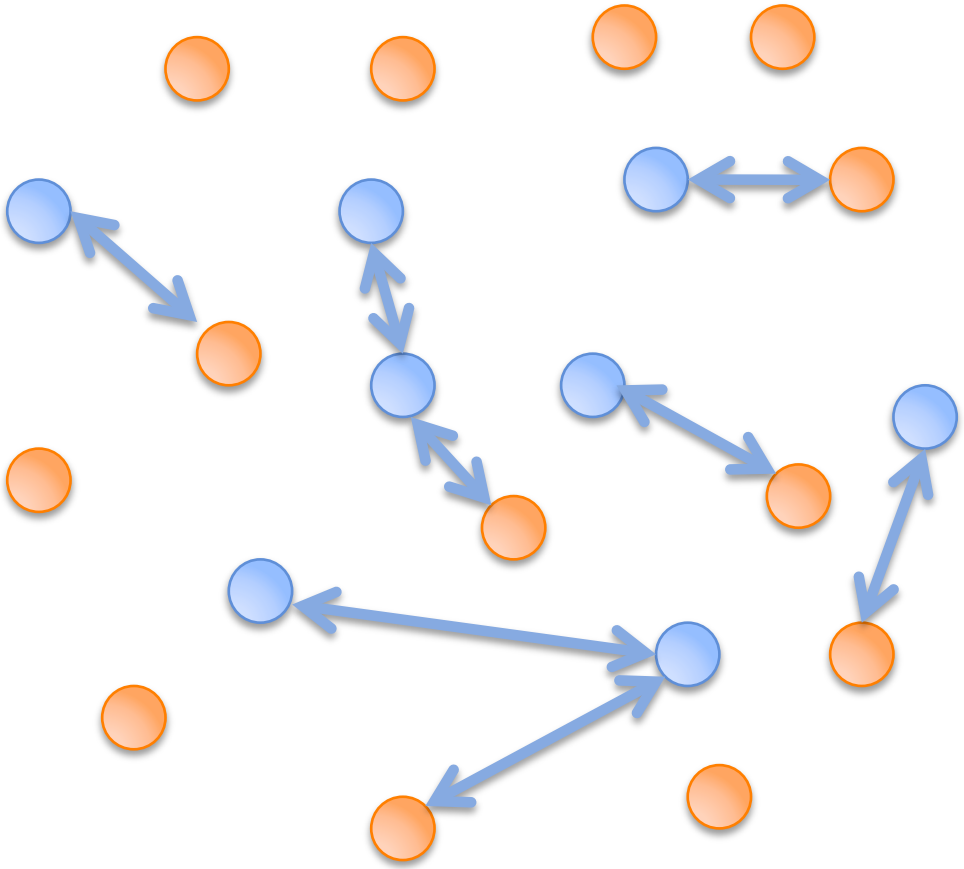
COMPLEX EPIDEMICS

- Sites are initially "susceptible"
- If a node receives an update("hot rumor"), it becomes "infective"
- Randomly choose a node and share the update
- If chosen node is "infective", the node becomes "removed" with probability 1/k
  - k=1, 20% miss the rumor
  - k=2, 6% miss the rumor
- Requires fewer resources
- There is a chance that an update will not reach all sites.

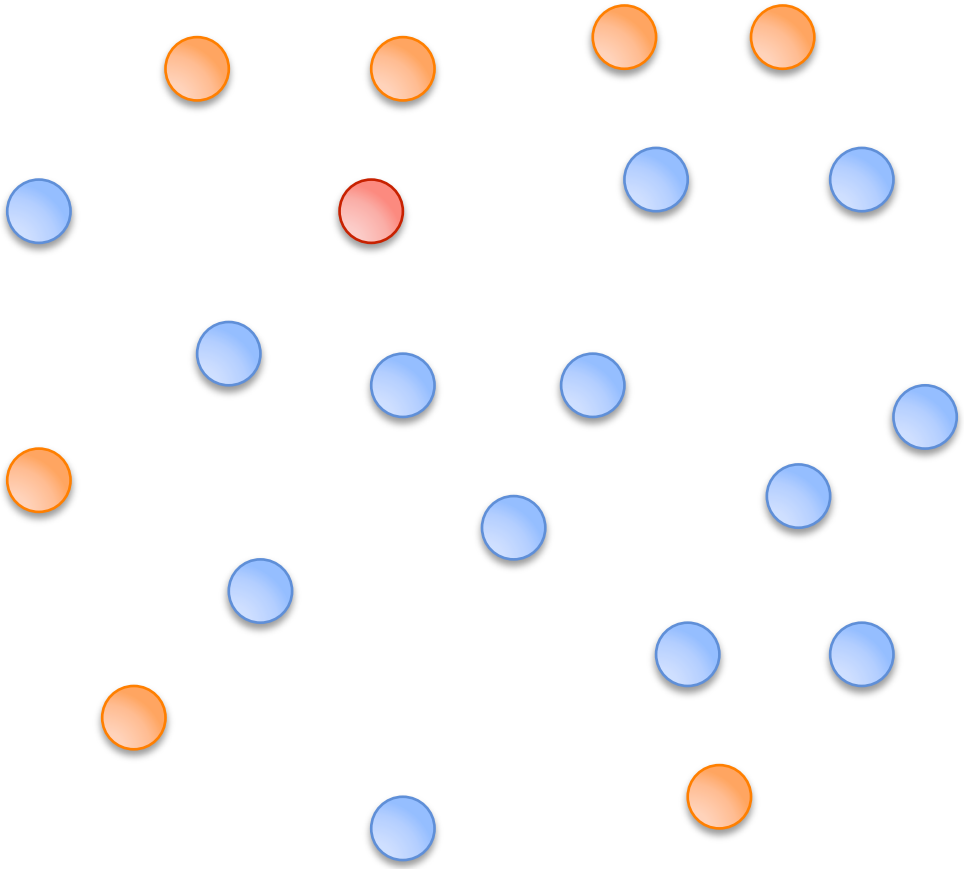
RUMOR MONGERING



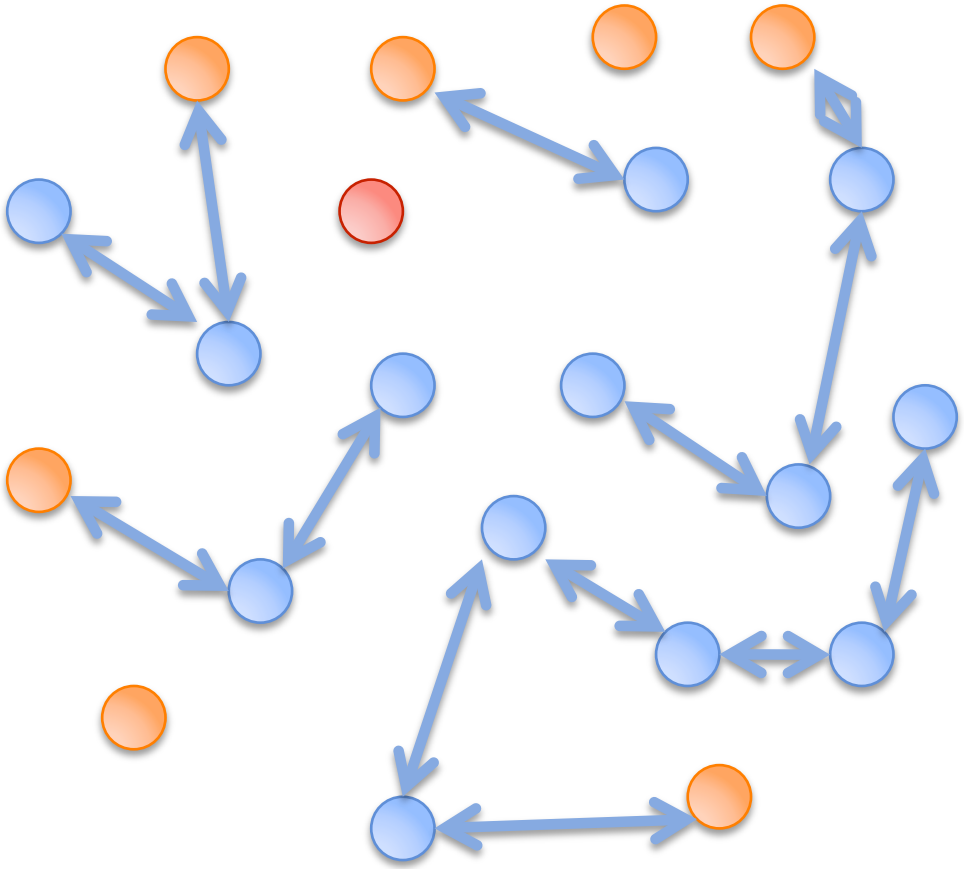
RUMOR MONGERING



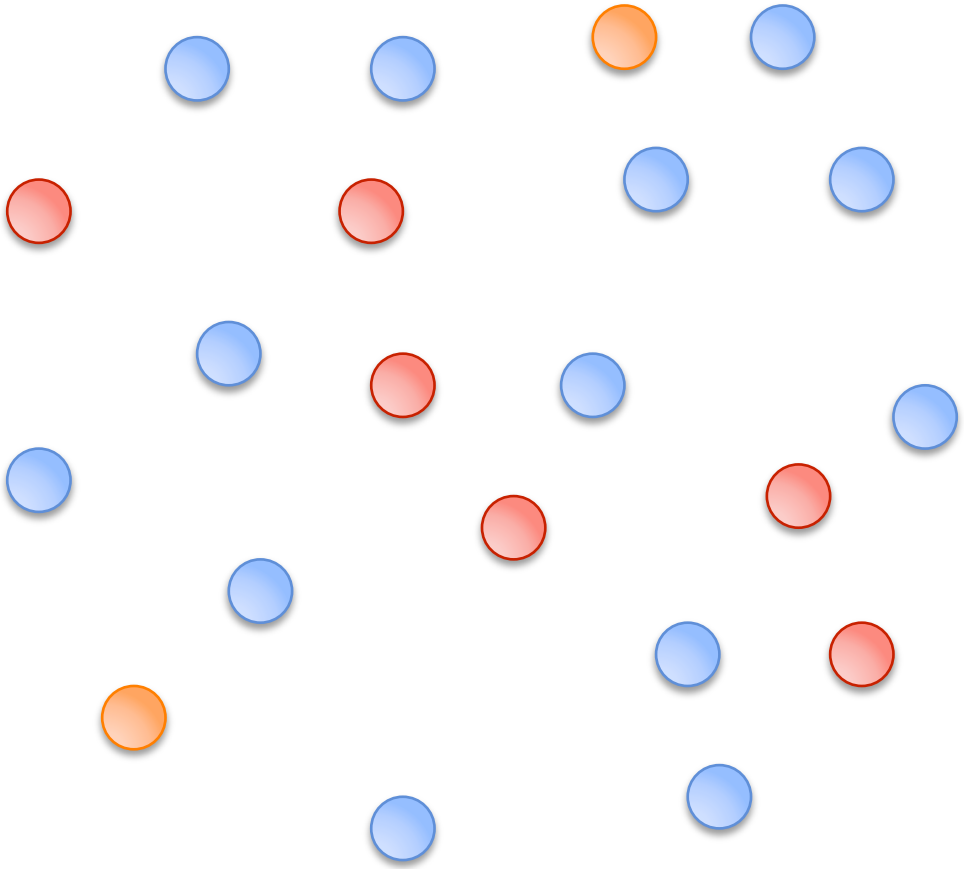
RUMOR MONGERING



RUMOR MONGERING



RUMOR MONGERING



VARIATIONS OF COMPLEX EPIDEMIC

- Criteria to judge epidemics
  - Residue: remaining susceptibles when the epidemic finishes
  - Traffic: Total update traffic / # of sites
  - Delay:
    - t_{avg} : Average of the difference between the time of the initial injection and the arrival of the update at a given site
    - t_{last} : the delay until the last site receives the update

VARIATIONS (CONT')

- Blind vs. Feedback
- Counter vs. Coin
- Push vs. Pull
- Minimization
- Connection Limit
- Hunting

Table 1. Push, Feedback & Counters

Counter	Residue	Trafic	Convergence	
k	s	m	t_{avg}	t_{last}
1	0.176	1.74	11.0	16.8
2	0.037	3.30	12.1	16.9
3	0.011	4.53	12.5	17.4
4	0.0036	5.64	12.7	17.5
5	0.0012	6.68	12.8	17.7

Table 2. Push, Blind & Coin

1	0.960	0.04	19	38
2	0.205	1.59	17	33
3	0.060	2.82	15	32
4	0.021	3.91	14.1	32
5	0.008	4.95	13.8	32

Table 3. Pull, Feedback & Counters

1	0.031	2.7	9.97	17.63
2	0.00058	4.49	10.07	15.39
3	0.000004	6.09	10.08	14.00

COMBINING ANTI-ENTROPY & COMPLEX EPIDEMIC

- Complex Epidemic
  - (+) spread updates rapidly
  - (+) low network traffic
  - (-) sometimes can fail
- Anti-entropy
  - (+) robust
  - (-) expensive
- Complex Epidemic for an initial distribution / redistribution and Anti-entropy for an infrequent backup
  - Every update eventually reaches every site
  - Can significantly reduce the cost of distribution

DELETION AND DEATH CERTIFICATES

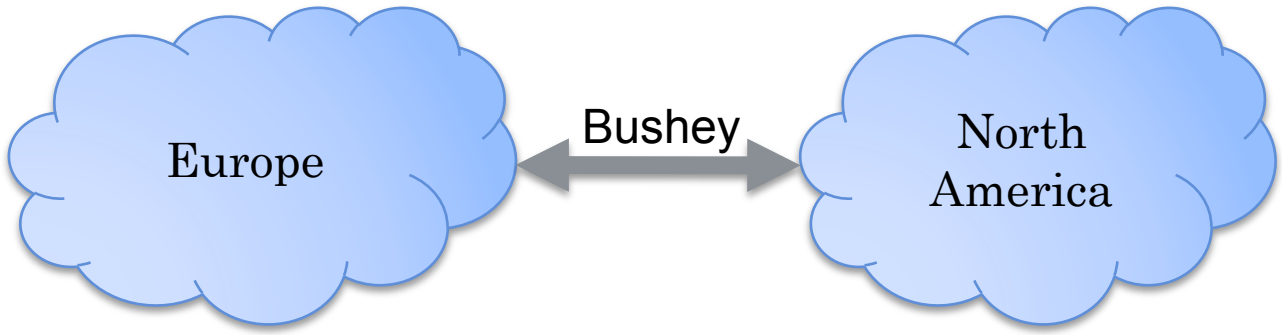
- Replace deleted items with *death certificates*(DC) which carry timestamps
- When old copies of deleted items meet death certificates, remove them
- Delete death certificates
  - When every site has seen it
  - After holding it for fixed threshold of time

DORMANT DEATH CERTIFICATES

- Tradeoff between the amount of space and the risk of obsolete items being resurrected
- After threshold *t1*, delete old DCs at most sites, retaining "dormant" at only a few cites
- When obsolete update encounters a dormant DC, propagate DC again
- After *t1*+*t2*, discard dormant DCs.

SPATIAL DISTRIBUTIONS

- Nonuniform spatial distribution can significantly reduce the traffic
  - Favor nearby neighbors
  - Trade off between convergence time and average traffic per link



CIN topology

SPATIAL DISTRIBUTION: ANTI-ENTROPY

- Can significantly reduce traffic on critical links
- Slow convergence doesn't change the total amount of traffic
- Robust: With probability 1, update converges.

Spatial	t_{last}	t_{avg}	Compare Traffic		Update Traffic	
Distribution			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

SPATIAL DISTRIBUTION:RUMOR MONGERING

- Less robust than anti-entropy
- Increase k to give 100% distribution
- Worthwhile improvements

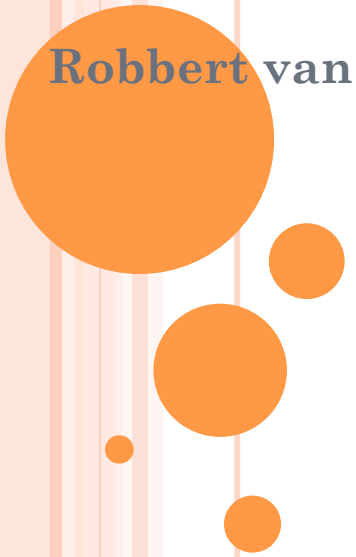
Spatial	k	t_{last}	t_{avg}	Compare Traffic		Update Traffic	
Distribution				Average	Bushey	Average	Bushey
uniform	4	7.83	5.32	8.87	114.0	5.84	75.87
a=1.2	6	10.14	6.33	3.20	18.0	2.60	17.25
a=1.4	5	10.27	6.31	2.86	13.0	2.49	14.05
a=1.6	8	11.24	6.90	2.94	9.80	2.27	10.54
a=1.8	7	12.04	7.24	2.40	5.91	2.08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

CONCLUSION

- Characteristics of Anti-entropy and Rumor Mongering
- Anti-entropy
  - can spread updates to all node with probability 1
  - Reduces average link traffic
- Rumor Mongering
  - If combined with anti-entropy, can greatly reduce the cost of redistribution
- Requires Membership though

ASTROLABE: A ROBUST AND SCALABLE TECHNOLOGY FOR DISTRIBUTED SYSTEM MONITORING, MANAGEMENT, AND DATA MINING

Robbert van Renesse, Kenneth P. Birman, and Werner Vogels



AUTHORS

Ken Birman
Cornell Univ.

Robbert van Renesse
Cornell Univ.

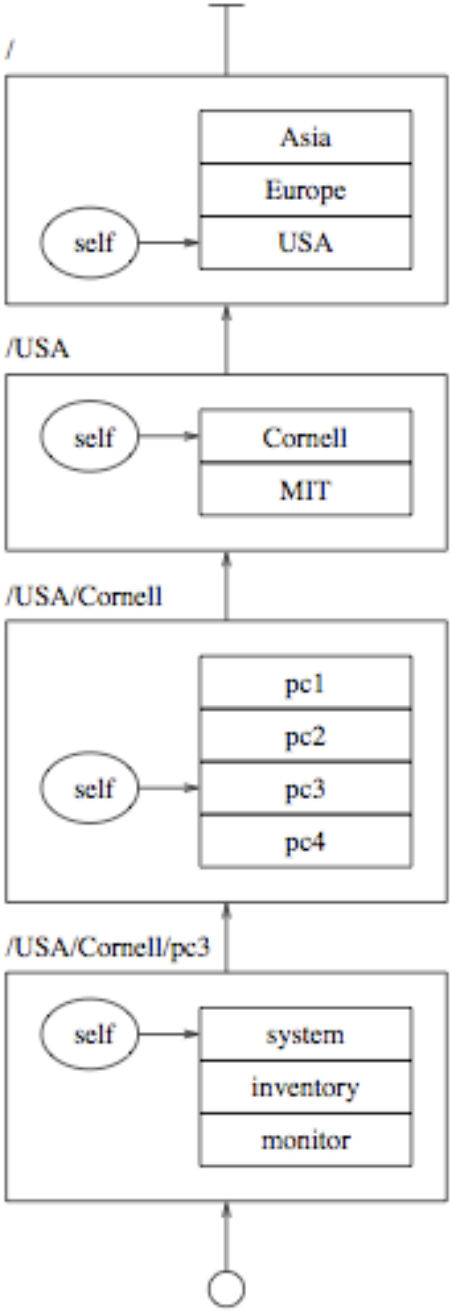
Werner Vogels
Amazon.com CTO

OVERVIEW

- Distributed Information Management System
  - Monitors the dynamically changing state of resources
  - Reports summaries of information
- Design principles
  - Scalability through hierarchy
  - Flexibility through mobile code
  - Robustness through a randomized p2p protocol
  - Security through certificates
- Now used in Amazon.com

ZONE AND MIB

- Zone name: path of zone identifiers from the root
- Management Information Base(MIB): attribute list containing the information associated with the zone
  - Each agent has a local copy of the root MIB, the MIBs of each child of the root
  - Each agent maintains MIB list of child zones



GOSSIP PROTOCOL

- Epidemic p2p protocol to propagate information
- Each zone elects a set of representative agents to gossip on behalf of that zone.
  - An agent may represent more than one zone
- Each agent periodically gossips
  - Picks one of the child zones at random,
  - Picks a random agent from child's list
  - Sends attributes of all the child zones up to root level.
- Gossip spreads quickly, O(log n)
- For scalability, robustness, and rapid dissemination of updates, eventual consistency is adopted.

EVENTUAL CONSISTENCY

- Probabilistic consistency
- Given an aggregate attribute X that depends on some other attribute Y
  - When an update u is made to Y, either u itself, or an update to Y made after u, is eventually reflected in X
  - With probability 1

ASTROLABE IS A FLEXIBLE MONITORING OVERLAY

swift.cs.cornell.edu

Name	Time	Load	Weblogic?	SMTP?	Word Version
swift	2271	1.8	0	1	6.2
falcon	1971	1.5	1	0	4.1
cardinal	2004	4.5	1	0	6.0

Periodically, pull data from monitored systems

cardinal.cs.cornell.edu

Name	Time	Load	Weblogic ?	SMTP?	Word Version
swift	2003	.67	0	1	6.2
falcon	1976	2.7	1	0	4.1
cardinal	2231	1.7	1	1	6.0

From Ken's slide http://www.cs.cornell.edu/courses/cs614/2006fa/Slides/Epidemics.ppt

STATE MERGE: CORE OF ASTROLABE EPIDEMIC

swift.cs.cornell.edu

Name	Time	Load	Weblogic?	SMTP?	Word Version
swift	2011	2.0	0	1	6.2
falcon	1971	1.5	1	0	4.1
cardinal	2004	4.5	1	0	6.0

cardinal.cs.cornell.edu

Name	Time	Load	Weblogic?	SMTP?	Word Version
swift	2003	.67	0	1	6.2
falcon	1976	2.7	1	0	4.1
cardinal	2201	3.5	1	1	6.0

From Ken's slide http://www.cs.cornell.edu/courses/cs614/2006fa/Slides/Epidemics.ppt

STATE MERGE: CORE OF ASTROLABE EPIDEMIC

swift.cs.cornell.edu

Name	Time	Load	Weblogic?	SMTP?	Word Version
swift	2011	2.0	0	1	6.2
falcon	1971	1.5	1	0	4.1
cardinal	2004	4.5	1	0	6.0

cardinal.cs.cornell.edu

Name	Time	Load	Weblogic?	SMTP?	Word Version
swift	2003	.67	0	1	6.2
falcon	1976	2.7	1	0	4.1
cardinal	2201	3.5	1	1	6.0

swift	2011	2.0
cardinal	2201	3.5

From Ken's slide http://www.cs.cornell.edu/courses/cs614/2006fa/Slides/Epidemics.ppt

STATE MERGE: CORE OF ASTROLABE EPIDEMIC

swift.cs.cornell.edu

Name	Time	Load	Weblogic?	SMTP?	Word Version
swift	2011	2.0	0	1	6.2
falcon	1971	1.5	1	0	4.1
cardinal	2201	3.5	1	0	6.0

cardinal.cs.cornell.edu

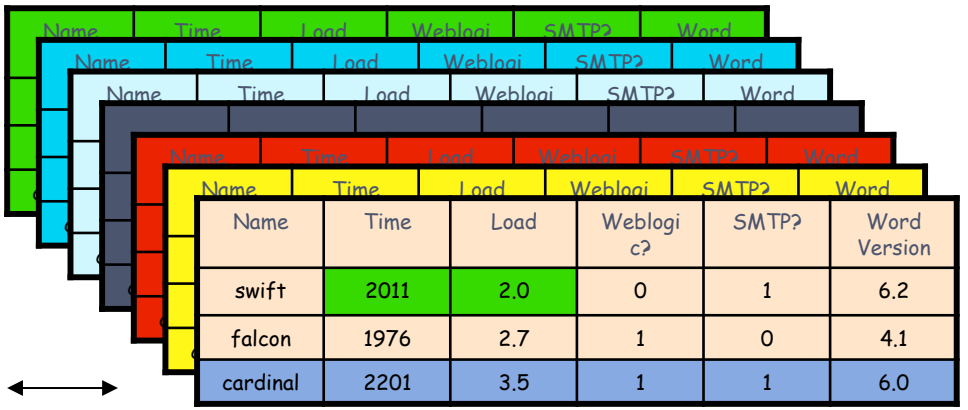
Name	Time	Load	Weblogic?	SMTP?	Word Version
swift	2011	2.0	0	1	6.2
falcon	1976	2.7	1	0	4.1
cardinal	2201	3.5	1	1	6.0

From Ken's slide http://www.cs.cornell.edu/courses/cs614/2006fa/Slides/Epidemics.ppt

SCALING UP… AND UP…

- We don't want every system to "see" all others (cost would be huge)
- Instead, structure into "zones". You only see data from your neighbors…

cardinal.cs.cornell.edu



Name	Time	Load	Weblogic?	SMTP?	Word Version
swift	2011	2.0	0	1	6.2
falcon	1976	2.7	1	0	4.1
cardinal	2201	3.5	1	1	6.0

From Ken's slide http://www.cs.cornell.edu/courses/cs614/2006fa/Slides/Epidemics.ppt

A FORM OF DATA MINING CONTINUOUSLY SUMMARIZES REMOTE INFORMATION

Dynamically changing query output is visible system-wide

Name	Avg Load	WL contact	SMTP contact
SF	2.6	123.45.61.3	123.45.61.17
NJ	1.8	127.16.77.6	127.16.77.11
Paris	3.1	14.66.71.8	14.66.71.12

SQL query "summarizes" data

Name	Load	Weblogic?	SMTP?	Word Version	…
swift	2.0	0	1	6.2	
falcon	1.5	1	0	4.1	
cardinal	4.5	1	0	6.0	

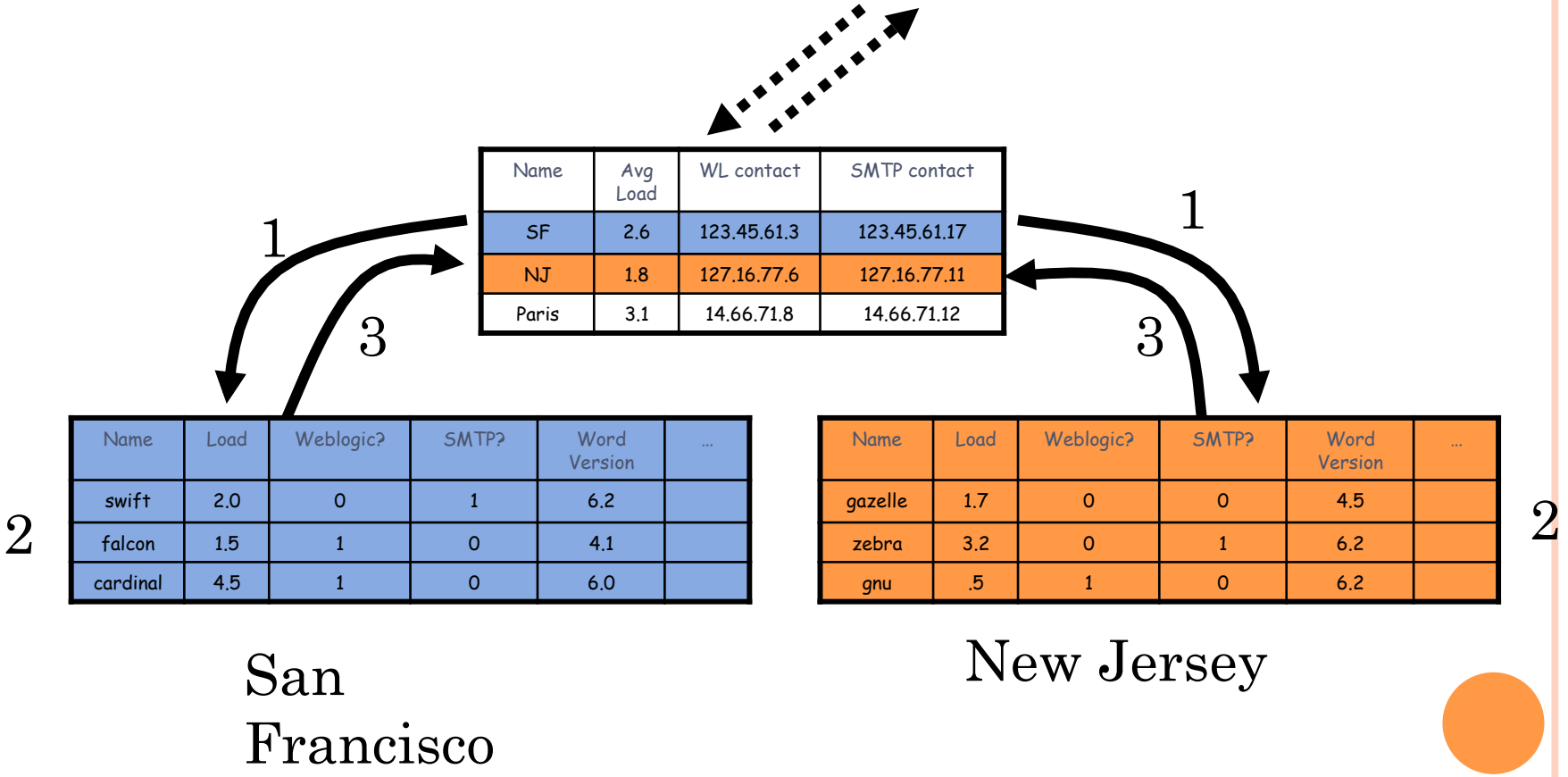
San Francisco

Name	Load	Weblogic?	SMTP?	Word Version	…
gazelle	1.7	0	0	4.5	
zebra	3.2	0	1	6.2	
gnu	.5	1	0	6.2	

New Jersey

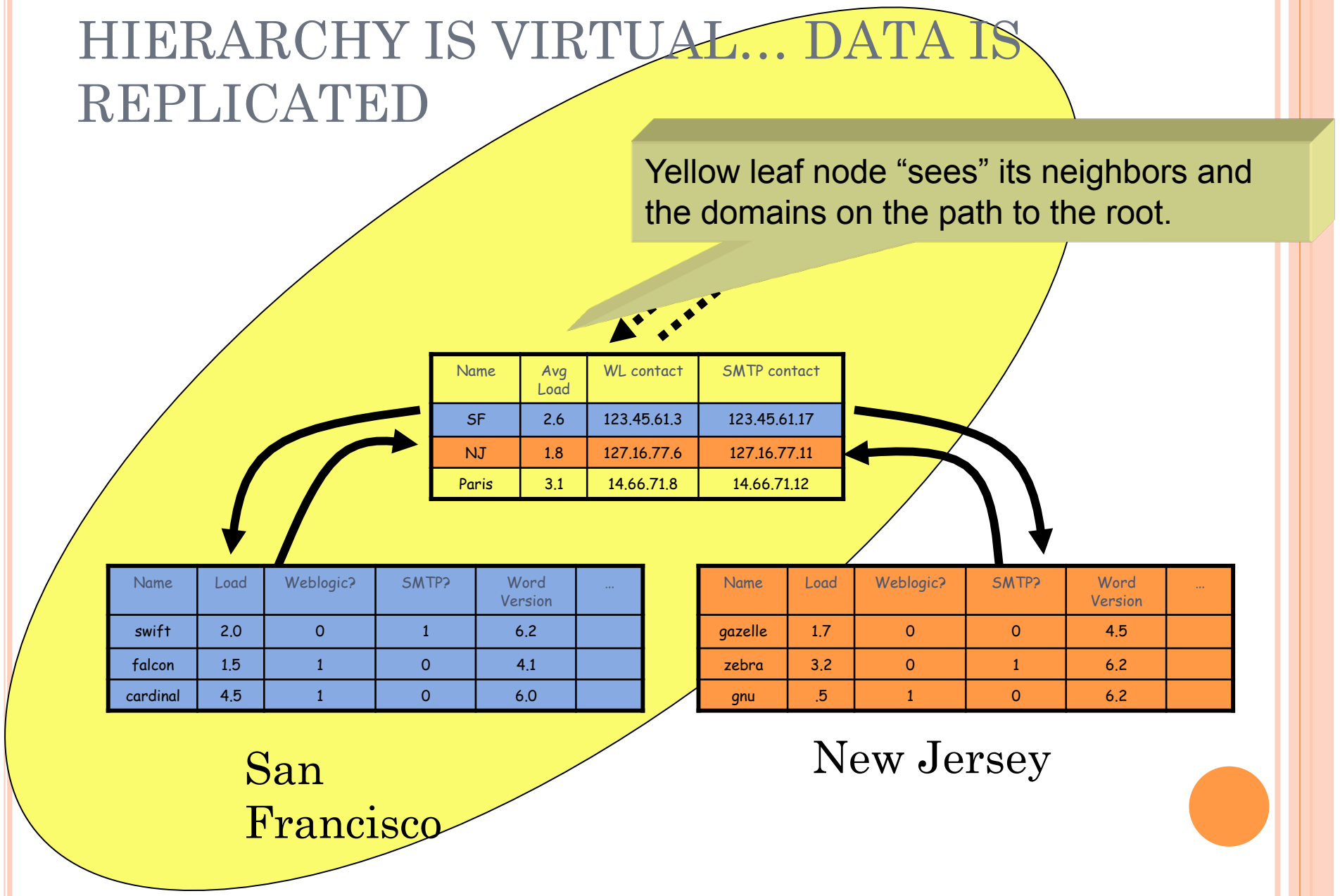
From Ken's slide http://www.cs.cornell.edu/courses/cs614/2006fa/Slides/Epidemics.ppt

(1) QUERY GOES OUT… (2) COMPUTE LOCALLY… (3) RESULTS FLOW TO TOP LEVEL OF THE HIERARCHY

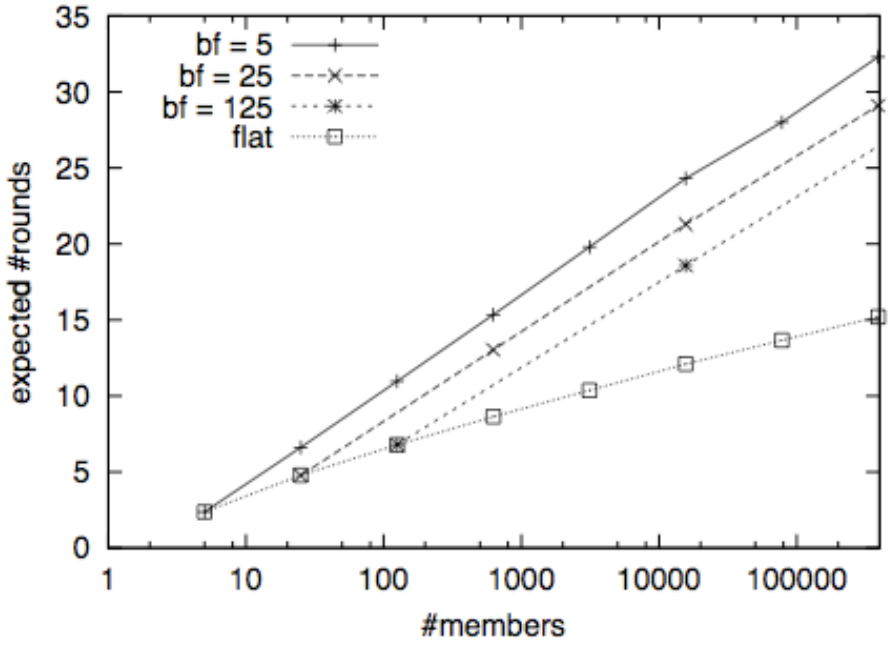


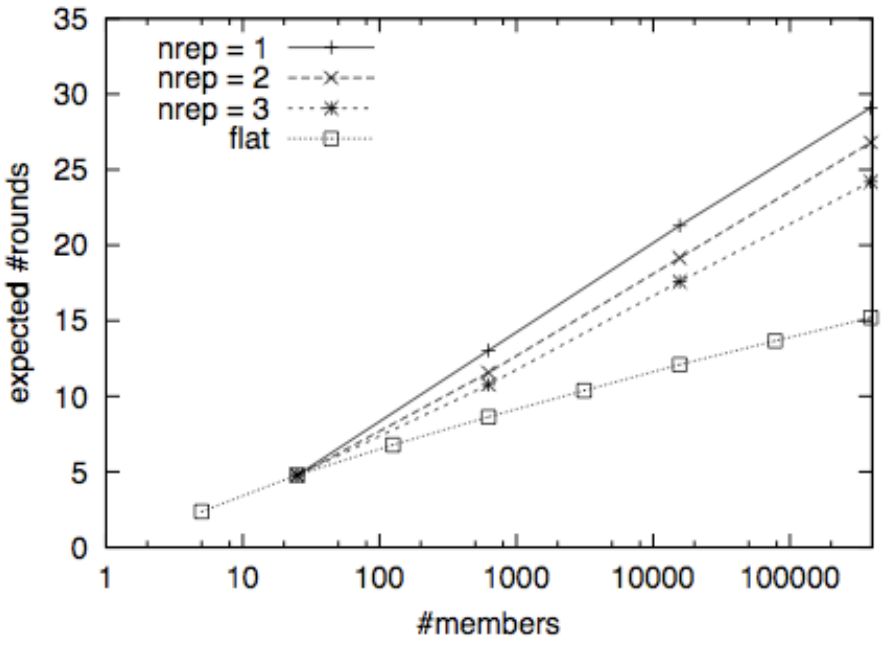
From Ken's slide http://www.cs.cornell.edu/courses/cs614/2006fa/Slides/Epidemics.ppt

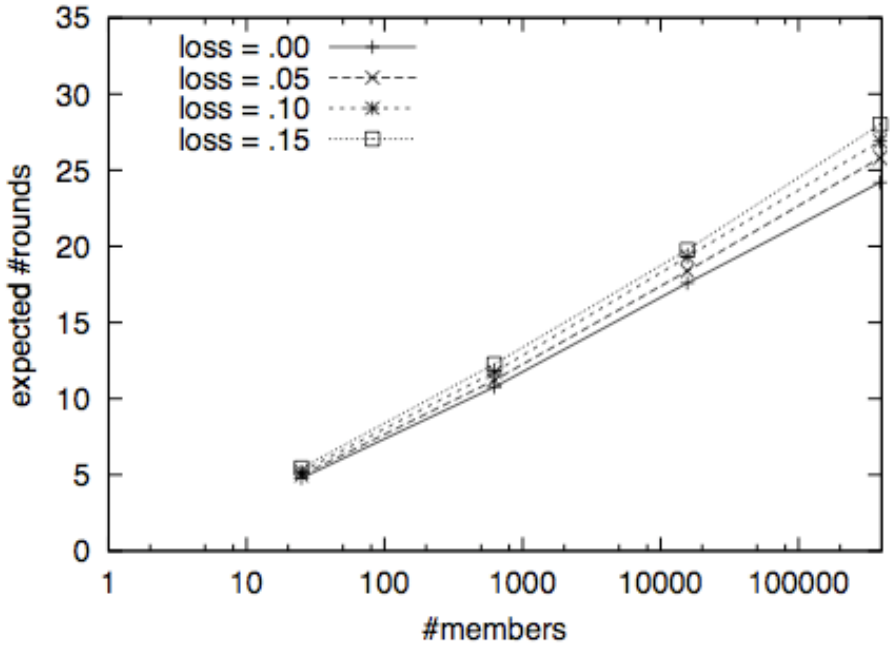
HIERARCHY IS VIRTUAL… DATA IS REPLICATED

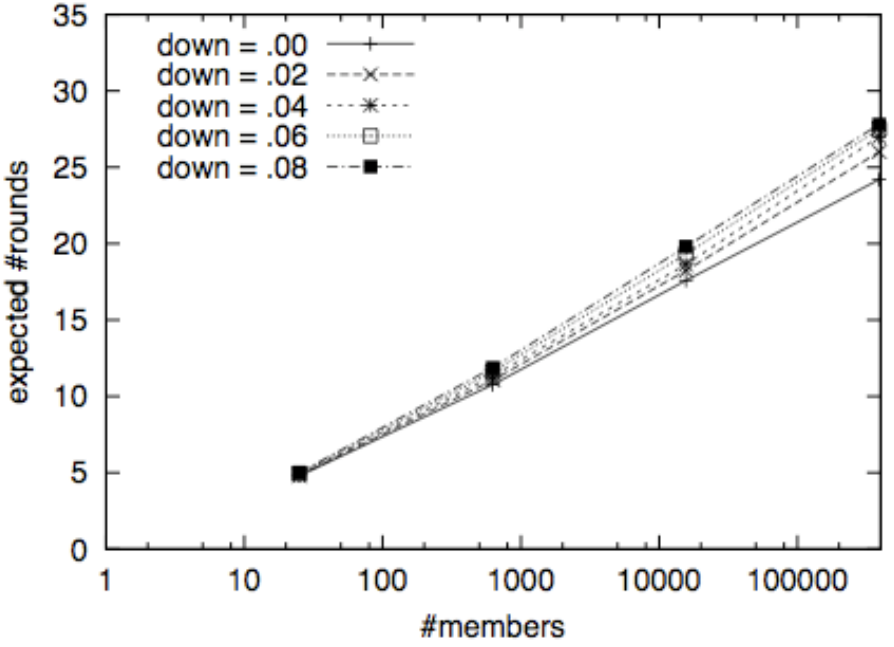


From Ken's slide http://www.cs.cornell.edu/courses/cs614/2006fa/Slides/Epidemics.ppt









Average # of rounds necessary to infect all nodes

CONCLUSION

- Tree-based gossip protocol
- Robust and scalable
- Eventual Consistency

DISCUSSION

- Anti-entropy vs. rumor mongering?
- Scalability?
- Eventual Consistency?

QUESTION?