Invited Talk:
Epidemic Protocols for Extreme-scale Computing

Global Knowledge without Global Communication

Dr. Giuseppe Di Fatta
G.DiFatta@reading.ac.uk
University of Reading

Wednesday, September 24, 2014
Outline

- **exXtreme-scale Computing**
  - Motivations: global knowledge without global communication
  - Applications: from distributed systems to exascale supercomputing (HPC)
    - Epidemic Data Mining

- **Epidemic Protocols**
  - Information dissemination and data aggregation
  - Membership and aggregation protocols

- **Open Issues and Contributions**
  - aggregation in asynchronous systems
  - local detection of global convergence
  - dynamics in overlay topologies

- **Conclusions**
From Large to Extreme-scale Systems

Distributed Systems

- Internet
  - Ubiquitous Computing, Crowd Sensing, P2P Overlay Networks
  - Internet of Things (50 to 100 trillion objects)
  - Decentralised Online Social Networks

- Ad-hoc Networks
  - Large-scale Wireless Sensor Networks
  - Mobile ad-hoc Networks (MANET)
  - Vehicular Ad-Hoc Networks (VANET)

Parallel Systems

- Towards exascale computing
  - Tianhe-2 (MilkyWay-2): National Supercomputer Center, Sun Yat-sen University, Guangzhou, China, Top500 N.1 since June 2013, 34/55 Pflop/s, 3.12M cores
Extremely Scalable Computing

- Scalability
  - number of data objects
  - dimensionality of data objects
  - number of processing elements

- Computing in extreme-scale systems
  - Scalability of the communication cost
  - Decentralisation
  - Robustness and fault-tolerance
  - Adaptiveness: ability to cope with dynamic environments

- Global Knowledge w/o Global Communication

G. Di Fatta
Epidemic Protocols

• A **communication and computation paradigm** for large-scale networked systems:
  – high scalability
  – probabilistic guarantees on convergence speed and accuracy
  – robustness, fault-tolerance, high stability under disruption

• aka Gossip-based protocols

*G. Di Fatta*
Exponential Growth

- In epidemiology, an epidemic is a disease outbreak that occurs when new cases exceed a "normal" expectation of propagation (a contained propagation).
  - The disease spreads person-to-person: the affected individuals become independent reservoirs leading to further exposures.
  - In uncontrolled outbreaks, there is an exponential growth of the infected cases.

Exponential Growth:
- In epidemiology, an epidemic is a disease outbreak that occurs when new cases exceed a "normal" expectation of propagation (a contained propagation).
  - The disease spreads person-to-person: the affected individuals become independent reservoirs leading to further exposures.
  - In uncontrolled outbreaks, there is an exponential growth of the infected cases.

Epidemic curve of the outbreak of new influenza A(H1N1) in Mexico and fitted exponential growth over the period 9 to 24 April 2009

Figure from: "Rapid communications A preliminary estimation of the reproduction ratio for new influenza A(H1N1) from the outbreak in Mexico, March-April 2009", P Y Boëlle, P Bernillon, J C Desenclos, Eurosurveillance, Volume 14, Issue 19, 14 May 2009


G. Di Fatta
Epidemic Computing

- Idea: Virus → Information

Disease outbreak

Epidemic communication for extreme-scale computing
Epidemic/Gossip-based Protocol

- A synchronous push mechanism for information dissemination (infection)
- Uniform Gossiping: assuming a node is able to select a node id (peer) uniformly at random
- Practical peer sampling: Membership Protocols are used to provide such a function in a practical way.

Active thread (cycle-based):
- Repeat
  - wait some $\Delta T$
  - chose a random peer
  - send local state

Passive thread (event-based):
- Repeat
  - receive remote state
  - If state==infected, then local state=infected
Information Dissemination: Propagation Time

- Time to propagate information originated at one peer

Time to complete “infection”: $O(\log N)$
Seminal Work and History

- Demers 1987 (Xerox PARC), Clearinghouse Directory Service
- Golding 1993, the refdbms distributed bibliographic database system
- Demers 1993-97 (Xerox PARC), the Bayou project
- Birman 1998 (Cornell), Bimodal Multicast
- van Renesse 1999 (Cornell), Astrolabe
- Karp 2000 (ICSI, Berkeley), Randomized Rumor Spreading
- In 2000-2005, a surge of studies:
  - several epidemic protocols and
  - their applications in communication networks and distributed systems
- Di Fatta 2011, first epidemic data mining algorithm for distributed systems
- Strakova 2011, first application to exascale supercomputing

- Theoretical work is still making progress but practical protocols and apps have received limited attention.

G. Di Fatta
Open Issues

• Theoretical studies and simulations typically assume
  - simplistic synchronous communication model with static/reliable network
  - unrealistic global knowledge of the networked system
  - the initial overlay topology is a random graph
  - unlimited or “enough” protocol rounds to reach convergence

• In distributed, large and extreme-scale networks:
  - communication is asynchronous, net is not reliable/is dynamic
  - nodes may only know a limited set of neighbours (sparse graph)
  - the initial topology may not be a random graph: poor initial topologies may have serious implications in convergence speed and, even worse, in the convergence guarantee itself
  - convergence is a global property that depends on several factors, which typically are not known locally.
Applications

• Epidemic protocols have been used to provide scalable and fault-tolerant services, such as:
  – information dissemination (broadcast, multicast)
  – data aggregation: values of aggregate functions more important than individual data (sum, average, sampling, percentiles, etc.)

• And they have been proposed for various applications:
  – DB replica synchronisation and maintenance
  – Network management and monitoring
  – Failure detection
  – HPC algs and services, e.g., QR factorization and power-capping
  – Epidemic Knowledge Discovery and Data Mining
    • decentralised discovery of global patterns and trends
Parallel K-Means in share-nothing systems

Global communication is not a feasible approach for extreme-scale systems.

- initialisation
  - generate centroids for first iteration

- distributed data
  - P0, P1, P2, P3

- distributed processes

- compute local clusters: partial sums

- broadcast

- centroids for next iteration: repeat until convergence

- data are intrinsically distributed

G. Di Fatta
Epidemic K-Means

- Distributed data
- Distributed processes

**Initialisation**
- Epidemic broadcast of a seed for the random number generator
- Generate centroids for first iteration
- Compute local clusters: partial sums

**Epidemic Aggregation**
- Of sums, counts and errors
- Centroids for next iteration: repeat until convergence

- Data are intrinsically distributed

P0

P1

P2

P3

(or static list of seeds for multiple executions)
Simulations - Data Distributions

- Each node has a fixed number of data points (100).
- Each data point belongs to a category (colour).
- Data points are assigned to nodes from uniformly at random (a) to locality-dependent allocation (d).

(a) $\lambda = 0$ ($J=0.9936$)

(b) $\lambda = 2$ ($J=0.4068$)

(c) $\lambda = 3$ ($J=0.2297$)

(d) $\lambda = 5$ ($J=0.1181$)
Clustering Accuracy

- Accuracy w.r.t. the “ideal” (centralised) data clustering

![Cluster distribution (Jain Index)](image1)

![Clustering Accuracy (average)](image2)

![Standard Deviation](image3)
Mean Squared Error of Centroids

- Error w.r.t. the “ideal” (centralised) centroids

Cluster distribution (Jain Index)

Clustering Error (average)

Standard Deviation
Fault-Tolerance of Epidemic K-Means

- Clustering accuracy under message loss and churn: 0-20%
The Data Aggregation Problem

• (a.k.a. the “node aggregation” problem)

• Given a network of N nodes, each node i holding a local value $x_i$, the goal is to determine the value of a global aggregation function $f()$ at every node:

$$f(x_0, x_1, ..., x_{N-1})$$

• Example of aggregation functions:
  – sum, average, max, min, random samples, quantiles and other aggregate databases queries.
Data Aggregation: e.g., Sum

- Centralised approach: all receive operations, and all additions, must be serialized: $O(N)$

- Divide-and-conquer strategy to perform the global sum with a binary tree: the number of communication steps is reduced from $O(N)$ to $O(\log(N))$.

\[
S = \sum_{i=0}^{N-1} x_i
\]
All-to-all Communication

- MPI AllReduce
  - MPI predefined operations: max, min, sum, product, and, or, xor
  - all processes compute identical results
  - number of communication steps: \( \log(N) \)
  - number of messages: \( N \log(N) \)

Any global function which can be approximated well using linear combinations.
Fault-Tolerance and Robustness

- The parallel approach requires global communication.
- It is not fault tolerant: even a single node or link failure cannot be tolerated.
- A delay on a single communication may have an effect on all nodes.
Epidemic/Gossip-based Protocol

Active thread (cycle-based):
• Repeat
  – wait some $\Delta T$
  – chose a random peer
  – send local state

Passive thread (event-based):
• Repeat
  – receive remote state
  – [reply with local state]
  – merge remote and local state

- definition of state and merge function (aggregation protocol)
  - E.g., average, sum, percentiles, etc.

- based on randomised communication: peer selection mechanism
  (membership protocol)
Epidemic Data Aggregation: Global Average

- Simulation of epidemic aggregation: local estimations of global average
- Network of 10K nodes: each node holds a local value.
  - Worst case analysis: peak distribution, i.e. information originated at one node

- Very high value
- Higher value
- Target value (0.01% error)
- Lower value
Epidemic Protocols

- **Push epidemic**
  - each peer sends state to other member

- **Pull epidemic**
  - each peer requests state from other member
  - expected #rounds the same

- **Push/Pull epidemic**
  - Push and Pull in one exchange
  - reduces #rounds at increased communication cost
Asymmetric/Symmetric Approaches

- in Uniform Gossip, at any cycle the probability that a node receives a number $x$ of messages follows a **binomial distribution**.
- **Asymmetric**: at each cycle, 36.8% of nodes do not receive any push message.
- **Symmetric**: at each cycle, every node receives at least one pull message.
The Push-Sum Protocol (PSP)

- Each node $i$ holds and updates the local sum $s_{t,i}$ and a weight $w_{t,i}$.
- Initialisation:
  - Node $i$ sends the pair $<x_i, w_{0,i}>$ to itself.
- At each cycle $t$:

  **Algorithm 1 Protocol Push-Sum**

  1. Let $\{(\hat{s}_r, \hat{w}_r)\}$ be all pairs sent to $i$ in round $t - 1$
  2. Let $s_{t,i} := \sum_r \hat{s}_r$, $w_{t,i} := \sum_r \hat{w}_r$
  3. Choose a target $f_t(i)$ uniformly at random
  4. Send the pair $(\frac{1}{2}s_{t,i}, \frac{1}{2}w_{t,i})$ to $f_t(i)$ and $i$ (yourself)
  5. $\frac{s_{t,i}}{w_{t,i}}$ is the estimate of the average in step $t$

- Update at node $i$:

  $s_{t+1,i} = \frac{1}{2}s_{t,j} + \frac{1}{2}s_{t,i} + \frac{1}{2}s_{t,z}$

  $w_{t+1,i} = \frac{1}{2}w_{t,j} + \frac{1}{2}w_{t,i} + \frac{1}{2}w_{t,z}$

  variance reduction step
The Push-Sum Protocol (PSP)

- **Settings** for various aggregation functions:

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sum</td>
<td>( v_i = ) local value ( w_i = 1 ) at a single node, 0 at all other nodes</td>
</tr>
<tr>
<td>Count</td>
<td>( v_i = 1 ) ( w_i = 1 ) at a single node, 0 at all other nodes</td>
</tr>
<tr>
<td>Average</td>
<td>( v_i = ) local value ( w_i = 1 )</td>
</tr>
<tr>
<td>Weighted Average</td>
<td>( v_i = ) local value \times local weight ( w_i = ) local weight</td>
</tr>
</tbody>
</table>

- **Convergence**: accuracy, consistency and speed
Mass Conservation Invariant

• The mass conservation invariant states that, for the case “Average”, the average of all local values is always the correct average and the sum of all weights is always $N$.

• Protocols violating this invariant do not converge to the true global aggregate.
Diffusion Speed

• **diffusion speed** is how quickly values originating at a source diffuse evenly through a network (convergence).
  
  – The number of protocol iterations such that the value at a node is diffused through the network, i.e., a peak distribution is transformed in a uniform distribution.
  
  – The diffusion speed is typically given as the complexity of the number of iteration steps as function of the network size, maximum error and maximum probability that the approximation at a node is larger than the maximum error.

• **PSP diffusion speed**: with probability $1 - \delta$ the relative error in the approximation of the global aggregate is within $\varepsilon$, in at most $O(\log(N) + \log(1/\varepsilon) + \log(1/\delta))$ cycles, where $\varepsilon$ and $\delta$ are arbitrarily small positive constants.
The Push-Pull Gossip (PPG) Protocol

- PPG aggregated average:
  - at a push msg nodes reply with a pull msg: local values are exchanged and averaged.
    - Node $i$ selects a random node $j$ to exchange their local values.
    - Each node compute the average and updates the local pair.
      - Variance reduction step: 
        \[
        v_{t+1,i} = \frac{1}{2}(v_{t,j} + v_{t,i}) \\
        v_{t+1,j} = \frac{1}{2}(v_{t,j} + v_{t,i})
        \]

- The push-pull operations need to be performed atomically.
  - If not, the conservation of mass in the system is not guaranteed and the protocol does not converge to the true global aggregate.
The Symmetric Push-Sum Protocol (SPSP)

- SPSP is a Push-Pull scheme with asynchronous communication
  - no atomic operation is required.

At each node $i$

**Require:** $v_0, w_0$
- The initial local value, $v_0$;
- The initial local weight, $w_0$.

**Initialisation:**
1. $(v, w) = (v_0, w_0)$

At each cycle:
2. $j \leftarrow \text{getNode}()$
3. $v \leftarrow v/2, w \leftarrow w/2$
4. send an aggregation message to $j$, $\langle (v, w), \text{true} \rangle$

At event: received an aggregation message $\langle (v', w'), r \rangle$ from $j$
5. if $r$ is true then
6. $v \leftarrow v/2, w \leftarrow w/2$
7. send an aggregation message to $j$, $\langle (v, w), \text{false} \rangle$
8. end if
9. $v \leftarrow v + v', w \leftarrow w + w'$

\[
\frac{1}{2} s_{t,i}, \frac{1}{2} w_{t,i} \quad \frac{1}{2} s_{t,j}, \frac{1}{2} w_{t,j} \quad \frac{1}{2} s_{t,j}, \frac{1}{2} w_{t,j} \quad \frac{1}{2} s_{t,i}, \frac{1}{2} w_{t,i}
\]
Comparative Analysis (PSP, PPG, SPSP)

- Convergence speed: variance of the estimated global aggregate over time
  - Percentage of operations with atomicity violation (AVP): 0.3% and 90%,
  - Internet-like topologies, 5000 nodes.
  - PPG and SPSP convergence speed is similar w.r.t. AVP.
Comparative Analysis (PSP, PPG, SPSP)

- The mean percentage error (MPE) over time
  - different AVP levels (from 0.3% to 90%)
  - averages over 100 different simulations: Internet-like and mesh topologies, 1000-5000 nodes, different data distributions, asynchronous communication.
  - Only PSP and SPSP converge to the true global aggregate value.
Epidemic Membership Protocols
## Protocol Stack

<table>
<thead>
<tr>
<th>Epidemic application</th>
<th>Epidemic application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggregation Protocol</td>
<td>Aggregation Protocol</td>
</tr>
<tr>
<td>Aggregation Protocol</td>
<td>Aggregation Protocol</td>
</tr>
</tbody>
</table>

**Membership Protocol**

- Transport Protocol
- Network Protocol

← Uniform Gossiping

← overlay topology

← physical topology
Epidemic & Membership Protocols

• Epidemic Protocols:
  – exchange information with other nodes to achieve some application goals (e.g., information dissemination, data aggregation)

• Membership Protocols:
  – provide the random peer sampling service for the above and is based on an epidemic approach too.

1. request a random node
2. response with random node j
3. send a push msg to j

node i

Epidemic Protocol

Membership Protocol

node j

Epidemic Protocol

G. Di Fatta
Overlay Topologies

- The overlay topology must have nice properties.
  - Sparse (out degree): e.g., a fully connected graph is not scalable (global knowledge)
  - Robust: no single points of failure - a star topology has optimal propagation time, but it is not scalable and is not robust.
  - Load balancing (indegree): there should not be bottlenecks.
  - Connectivity: a single connected component
    - The overlay topology must be connected at all times. If at any time the graph degenerates into multiple connected components, it will not heal (*) and the application-layer epidemic protocol will not converge.
  - Good propagation/diffusion: random graphs, expanders

(*) with current protocols
Epidemic Membership Protocols

• Practical peer sampling:
  – Partial view of the global system: a local cache of (max size) peer IDs is maintained and used to draw a random entry when requested
    • The cache is initialised with the initial (physical) neighbours.
    • Caches are periodically exchanged (likewise push/pull messages), merged and randomly trimmed to max size.
  – This is equivalent to multiple random walks: the cache entries quickly converges to a random sample of the peers with uniform distribution.
    • random sparse regular graph

\[
\begin{array}{c}
\text{node i} \\
\begin{array}{c}
a \\
b \\
j \\
c \\
\end{array}
\end{array}
\stackrel{\text{Membership Protocol}}{\longrightarrow}
\begin{array}{c}
\text{node j} \\
\begin{array}{c}
d \\
e \\
f \\
g \\
\end{array}
\end{array}
\]

G. Di Fatta
Epidemic Membership Protocols

• Practical peer sampling:
  – Partial view of the global system: a local cache of (max size) peer IDs is maintained and used to draw a random entry when requested
    • The cache is initialised with the initial (physical) neighbours.
    • Caches are periodically exchanged (likewise push/pull messages), merged and randomly trimmed to max size.
  – This is equivalent to multiple random walks: the cache entries quickly converges to a random sample of the peers with uniform distribution.
    • random sparse regular graph

G. Di Fatta
Epidemic Membership Protocols

- At each cycle (synchronous model), the distributed set of caches define a \textit{transient random overlay network}.
  - The membership protocol keeps changing the overlay topology over time.
  - Aim: the random node sampling from the local partial view results in a uniform distribution over the global system $\rightarrow$ Uniform Gossiping.
  - The node caches define an overlay network topology:
    - a random sparse regular digraph that keeps changing over time.

Some membership protocols:
- Node Cache Protocol, Cyclon, Send&Forget, Newscast, Eddy
Random Overlay Topologies

- A Membership Protocol is a fully-decentralised generative graph method:
  - it takes an input graph and generates a random output graph with similar properties.
    (assuming a simplified synchronous network model)
- Most (not all) MPs adopt (random) regular digraphs: the local cache has fixed size.
- Ideally we would like the MP to induce an attraction towards strongly connected graphs with equal indegree (or with low variance): the indegree can be used as a measure of robustness.

![Diagram showing different types of digraphs and their connections](image)

- Initial condition
The Expander Membership Protocol

• A novel membership protocols inspired by the concept of expander graphs, aka ‘expanders’.

• An expander is a sparse graph with strong connectivity.
  – The strong connectivity can be quantified by an index of expansion quality.

• The Expander Membership Protocol is designed to maximise the expansion quality of the overlay topology.
  – quasi-random peer selection: random search of a push-pull peer that minimizes cache overlap.
Expansion Quality

- The vertex expansion index $h(V, S)$ and its minimum over different sample sizes (typically $0 < s < \frac{1}{2} |V|$):

$$h(V, S) = \frac{|\partial(S)|}{|V \setminus S|}$$

$$h_{\text{min}}(V, s) = \min_{S \subseteq V, |S|=s} \frac{|\partial(S)|}{|V \setminus S|}$$

$V$: the set of network nodes

$S$: a sample of nodes, $S \subset V$, $|S|=s$

$\partial(S)$: the boundary of $S$, i.e. the set of nodes not in $S$ and 1-hop distant from at least one node in $S$. 

![Diagram showing the relationship between the set of network nodes, a sample set, and their boundary](image)
Message Forwarding Mechanism

• Case 1:
  – $Q_x$ is local cache of node $x$ and $Q_y$ is local cache of node $y$.
  – Each iteration node $x$ will send push message to node $y$.
  – If $|Q_x \cap Q_y| \leq T_{\text{max}}$, then $y$ will accept the push message and reply with pull message.
Message Forwarding Mechanism

- **Case 2:**
  - Each iteration node $x$ will send push message to randomly selected node $y$.
  - If $|Q_x \cap Q_y| > T_{\text{max}}$, then $y$ will forward the push message to another randomly selected node from $Q_y$ and repeat the same step until the message is accepted.
Message Forwarding Mechanism

- **Case 3:**
  - In order to prevent excessive communication overhead and delay, the forwarding procedure will be repeated up to $H_{\text{max}}$, then the message will return to the node with lowest similarity and force it to accept the message.
Recovery Mechanism (WIP)

• The simple protocol may still lead to multiple connected components when
  – the initial condition is particularly poor (e.g., ring of communities) and
  – in the presence of interleaving in push-pull operations

• An additional heuristic mechanism has been incorporated to facilitate the recovery from multiple connected components back to a single one.
  – Work in progress: only limited experimental verification

• General idea to fix loss of connectivity:
  – Interleaving causes unwanted duplication of cache entries
    • If somewhere a duplicate is generated, somewhere else an unwanted drop is made.
  – Some selected entries rather than dropped are stored in a secondary cache.
  – When local duplication is detected, then entries from secondary cache is recovered.
Simulations

- Task: computing global aggregation value (peak distribution)
- Network size: 10000
- Aggregation protocol: SPSP, peak distribution
- Initial overlay topologies (with poor expansion):

  - circular regular graph
  - ring of communities
Minimum Expansion Index

• Comparison of different membership protocols
  – init: circular regular graph
  – chart: minimum vertex expansion index: $h_{\text{min}}(G,5\%|V|)$
Minimum Expansion Index

- Comparison of different membership protocols
  - init: ring of communities
  - chart: minimum vertex expansion index: $h_{\text{min}}(G,5\%|V|)$
Convergence Speed

- Comparison of different membership protocols
  - init: circular regular graph
  - chart: convergence speed
Connected Components

- Comparison of different membership protocols
  - init: ring of communities
  - chart: max number (over several trials) of connected components vs #cycle

Max over MultiSeeds of # of Connected Com. RingCommunity, N=10000, Community Size = 1000

G. Di Fatta
Global Synchronisation
Convergence

1. Local convergence
2. Global convergence
3. Local detection of global convergence (global synchronisation)
   - Simulations:
     - 10K nodes, peak distribution, 5 Aggregation protocols, init random graph
     - Chart: number of nodes (%) locally converged to the global aggregate within a tolerance error for different accuracy thresholds (stddev).

G. Di Fatta

➤ based on global knowledge only available in simulations.
Global Synchronisation

• Can global convergence be estimated locally?
• Multiple independent aggregation protocols
  – local variance is used to detect convergence → **global synchronisation without global communication**

<table>
<thead>
<tr>
<th>Aggregation Protocol #1</th>
<th>Aggregation Protocol #2</th>
<th>…</th>
<th>Aggregation Protocol #k</th>
</tr>
</thead>
</table>

**Global Synchronisation**

\[
\{f_1(), f_2(), \ldots, f_k()\} \rightarrow [\mu, \sigma]
\]
Global Synchronisation

- Global convergence depends on several factors, network conditions and application requirements.
- Ideal synchronisation vs local detection method 6M

(a) varying $\varepsilon$ ($N = 10^4$)

(b) varying $N$ ($\varepsilon = 10^{-4}$)

- The ideal step transition (in red) is based on global knowledge, only available in simulations.
- The local method is based on local knowledge available at each node.
Global Synchronisation

- Convergence transition of different methods ($\varepsilon = 10^{-4}$ and $N = 10^4$)

(a) multiple protocols (M)

(b) History buffer (H)

(c) Multiple protocols with buffer (MH)

(d) comparison
Transition Period

- Nodes detect global convergence at different times during a transition period.
- Chart: the number of cycles from 0% to 100% of nodes that have detected global convergence for different methods.
Conclusions

• Extreme Computing based on Epidemic Protocols
  – fully decentralised
  – fault-tolerant
  – suitable for extreme-scale networked systems
  – suitable for asynchronous and dynamic networks

• Contributions:
  – Symmetric Push-Sum Protocols (SPSP), an aggregation protocol
  – The Expander Membership Protocol
  – Methods of global convergence detection (synchronisation)
  – Epidemic K-Means, the first epidemic data mining algorithm

• Current work
  – Refining and extending the Expander Membership Protocol: incorporating a connectivity recovery mechanism
Open Issues and Future Work

- Local estimation of convergence: better and faster convergence detection and synchronisation
- Asynchronous epidemic protocols (w/o cycles)
- Epidemic formulation of data mining algorithms: e.g., decision tree induction, recommender systems, etc.
- Protection against malicious nodes and loss of network connectivity

➤ Practical applicability still to be shown
  - Need to identify potential user applications and their deployment strategy


References

• Mathematical models of Epidemics
  – Nicholas C. Grassly & Christophe Fraser, "Mathematical models of infectious disease transmission, Nature Reviews Microbiology 6, 477-487 (June 2008)

• Gossip-based protocols for information dissemination:

• Gossip protocols for the data aggregation problem:

• Gossip-based protocols surveys, general studies, applications:
Questions?