On Maximizing The Lifetime Of Distributed Information In Ad-Hoc Networks With Individual Constraints

Cédric Westphal
Nokia Research Center
313 Fairchild Dr. Mountain View, CA94043
USA
cedric.westphal@nokia.com

ABSTRACT

Ad-Hoc Networks and in particular sensor networks are networks of nodes with limited battery and limited processing capacity. As such, a single node carries an incentive to limit the amount of data it contains. This leads to the expiration of the data carried by the node after a period of time, due for instance, to a re-boot after an off period in the duty cycle, or to older information being "pushed" out by new data received by the mobile node. On the other hand, some data is critical to the functioning of the whole network. For instance, the existence and position of a gateway towards the infrastructure network should be kept *somewhere* in the network, so that nodes are able to recover this information when needed. In this paper, we study the trade-off between the finite lifetime of a piece of information at each node, and the survival of this information indefinitely within the network. We consider a simple dissemination process for the information akin to an AODV-based information request/reply mechanism. We show that the maximum number of hops in a request is a critical parameter to ensure the survivability indefinitely of any information within the network. We identify the parameter which minimizes the load on the network, for two typical ad-hoc network topologies: a square lattice, which accurately models the distribution of the nodes in a fixed and organized ad hoc or sensor network, and a n-ary tree, which models ad hoc networks for which routing is constructed so as to have no routing loops.

Categories and Subject Descriptors

G.3 [**Probability and Statistics**]: Stochastic Processes;; C.4 [**Performance of Systems**]: Modeling Techniques;; H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Performance, Theory

MobiHoc'05, May 25–27, 2005, Urbana-Champaign, Illinois, USA.

Keywords

Ad-hoc networks, sensor networks, contact process, peer-to-peer systems.

1. INTRODUCTION

Ad-hoc networks are collection of nodes organized without the assistance of an infrastructure. The nodes join and leave the network, or move about, making it a permanent challenge to identify which node is able to provide which information to the other nodes in the network.

Further, resource constraints on the node limit the amount of data carried by each node. Due to these resource constraints, the data carried by the node disappears, or expires after some time. For instance, in a sensor network, a node might alternate between on and off states to economize battery power. At every wake-up time in the duty cycle, the states that were maintained in the previous cycle might have been flushed out. The node then starts the new cycle with a blank slate. The lifetime of the data which composed these states is thus that of the new *on* cycle.

In another example, the constrained resource could be the memory storage itself: a node in a ad hoc network might store data that keeps streaming in. For instance, a set of nodes sharing mp3 or mpeg files in an ad-hoc peer-to-peer network: the capacity of the typical portable multimedia player is much less than the size of a typical music library, let alone a movie library. After some initial period filling in the memory, some choice has to be made as to what to keep and what to discard. If all the data has the same priority, then a FIFO policy could be used to discard the oldest data and replace it with the newest. Other policies would yield different length of stay for the data on the device, but in most case, a finite lifetime is assigned to the data by the user's policies and resource constraints.

In an ad-hoc networks, without even considering that nodes could move relative to each other, locating data objects which has a finite lifetime might prove difficult: the data object that existed at time t might have expired at time $t + \tau$. Since each single piece of data will eventually disappear, a given object must be replicated throughout the network. For instance, a p2p network will contain multiple copies of the same mp3 file. Multiple sensor nodes in a wireless sensor network will contain a network configuration file pointing to the physical location of the information sinks in the sensor network, or some routing information, or some other data object pertaining to the function of the network.

While each node tends to discard data objects to satisfy its resource constraints, the network as a whole has an incentive to ensures that copies of each object are alive and distributed throughout the network. The object might be necessary for the network to work properly, as in the configuration file example in a sensor network. Or the wide availability of even the rarest objects might be an incentive for nodes to join the network, as in the peer-to-peer example. The network incentive is thus to preserve the existence of data objects while each single node has the resource constraint to remove such objects.

1.1 Motivation

The issue we are interested in becomes dual: first, is it possible to ensure that *live* replicas of a given object exist within the network at all times? And second, when a node requires the use of a data object, is it possible to locate such object efficiently?

Before stating the main results, we consider the problem of locating objects distributed throughout the network. A node needs to access a given object. In an ad hoc network, this first node is unable to refer to a centralized location registry, as there is no such infrastructure. The simplest way is for this node to broadcast an object request. The request is sent to its neighbors, who relay it to their neighbors, etc., until it finds a second node that has a live copy of the object. This second node can then reply to the first one with a copy of the object. This is the principle that AODV [8] uses to locate the correspondent of a node for routing purpose. It has been suggested [5] to extend this protocol to locate a more general class of objects than routing data. We consider a similar request/reply mechanism for the problem of locating objects in the ad hoc network.

More sophisticated solutions using distributed hash tables (see for instance [9, 10] and numerous others) exist, but we do not consider them at this point: since the lifetime of the object is not known *a priori* but is rather a parameter of the node holding the object, it is difficult for a DHT to always point to a live copy. If the DHT does not point to a live copy, then the broadcast mechanisms becomes the default mode anyway.

Of course, flooding the network will find any copy of the requested object. However, we would like to minimize the impact of the broadcasting phase, mostly by limiting the broadcast radius. As the number of nodes grows, the amount of broadcast traffic grows as well. Mechanisms exist to reduce the amount of data (minimal spanning tree broadcasting mechanism, such as [1] for instance) but they add on processing complexity.

The method we use to model this is the theory of contact processes: a contact process is a form of percolation in which nodes exchange information (or historically, diseases). The node possesses the information (is contaminated by the disease) for some random period of time. It transmits the information (disease) to its neighbors. At the core of the contact process is the rate of transmission vs. the rate of expiration or peremption of the information (transmission vs. cure of the disease): there exists a critical rate to ensure that the information (disease) stays alive in the system. From this rate depends also the density of the process, i.e., the likelihood that a node possesses the information (is contaminated). From this density we can infer the radius of a broadcast mechanism, for instance. We will use the terms infection or contamination, meaning that an infected node would reply positively to an object request.

The process we consider is different from the contact process: similarly, an object is kept at a node for a random period of time. But the dissemination of the replicas follows a different rule: a node which does not have the object will not receive it from its neighbor. It has to request the object, and will request it from the nodes within its broadcast radius. The request rate is an intrinsic parameter of the node, namely how often it needs the object. It is not a parameter dependent on the number of infected neighbors. We call our process the ad hoc dissemination process, which we will detail in Section 3. We will provide a brief background on the contact process in Section 2.

Figure 1 depicts a network which was uniformly seeded with replicas of one object. It shows the distribution of the objects after running the ad hoc dissemination process: nodes with a thicker diamond mark are the nodes which store the valid object, nodes with a light mark do not maintain a state with the object. However, the object is distributed somewhat uniformly in this scenario, and it is available to most nodes within 3 hops.

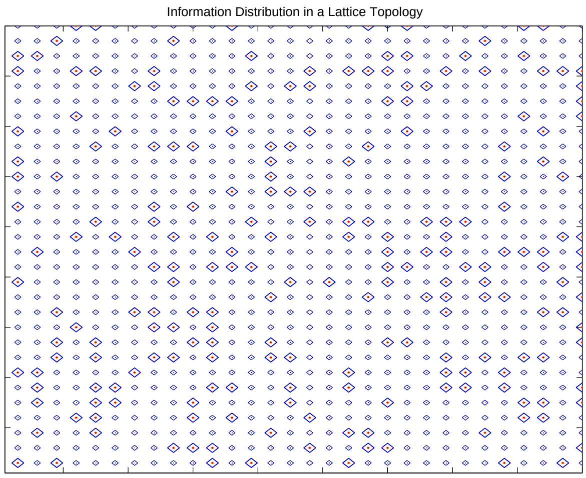

Figure 1: Information distribution over a lattice

1.2 Contribution

The main contribution of the paper is that it is the first study of a ad hoc network that stores object with a finite lifetime. The typical assumption in distributed systems is that the information is not perishable, something that is not true due to the individual nodes constraints in a ad-hoc or a sensor network. The main result is that for any object request rate, and for any object expiration rate, then the ad hoc dissemination process will survive, provided that the diameter of the request broadcast is chosen appropriately. This offers a value for the broadcast radius that minimizes the traffic for this category of object location mechanisms. We also believe that it is the first attempt at generalizing the contact process to the field of ad hoc networks, where we believe it would find a wide applicability. It has been used in a communication context, but in one dimension by [11].

We prove these results for two topologies: a lattice and a n-ary tree. We chose these topologies for a few reasons. The main reason is that we are able to complete the proofs. The other reason is that these topologies are good approximations of actual network topologies: a sensor network might be deployed by placing the nodes according to a lattice to ensure proper coverage of a given area. A Manhattan-like city coverage for example could be achieved by having a sensor at each street-and-avenue intersection. The tree topology arises naturally from the routing in a sensor network: the sink is the root of the tree, and each nodes has one root neighbor, and a few leaf neighbors. A tree topology could be imposed on an ad-hoc network to create a minimal spanning tree. For these reasons, these topologies are good first steps to provide meaningful results.

We already presented Section 2 and 3. In Section 4, we prove the existence of a critical rate for the ad hoc dissemination process. Above a certain rate for object requests, the process can survive with some positive probability. In section 5, we look at the search radius that is required if the rate request is not a varying parameter, but rather a static quantity. We use interchangeably the terms search depth or search radius to depict the number of hops we allow a request to travel. In section 6, we illustrate the results with some simulations and assess the critical rate value from the numerical results. We eventually conclude and give some directions for future work. All results and simulations are given both for a lattice topology and a n-ary tree topology.

2. PERCOLATION MODEL: THE CONTACT PROCESS

The contact process was first described by Harris [4]. For an in depth description of the contact process, we refer the reader to [6]. A variant similar to ours has been described in [3], in a different context and for different results. The contact process is a Markov process on the lattice. We restrict ourselves here to two dimensions, but Harris described the process in d dimensions. In 2 dimensions, the process $\{X_t^{cp}(v)\}$ takes value 1 if the node v with coordinates (x, y) possesses some object, or 0 otherwise. We will write that a node v is infected if $X(v) = 1$. We will use the term infected both in the contact process context, and in the ad hoc dissemination context. The value of $X_t^{cp}(v)$ depends on the value of X^{cp} for the neighbors of (x, y). A neighbor is a node connected directly to (x, y) in the lattice.

The initial condition of the process is the set S of nodes such that, at time 0, $X_0^{cp}(s) = 1$ if $s \in S$ and 0 otherwise.

The value of $X_t^{cp}(x, y)$ changes according to the following rules:

- if $X_t^{cp}(x, y) = 1$, then $X_t^{cp}(x, y)$ transitions to 0 as a Poisson process with rate 1.
- if $X_t^{cp}(x, y) = 0$ then $X_t^{cp}(x, y)$ transitions to 1 as a Poisson process with rate $\lambda|A_t(x, y)|$, where $A_t(x, y)$ is the set of neighbors w of (x, y) for which $X_t(w) = 1$.

One quantity of interest is the extinction time for the contact process when the initial condition at time 0 is S as it depends on λ, namely $\tau_S^{cp}(\lambda)$:

$$\tau_S^{cp}(\lambda) = \inf\{t : \forall x, y \in \mathbf{Z}^2, X_t^{cp}(x, y) = 0\} \quad (1)$$

One canonical result is that there exists a value of $0 < \lambda_c^{cp} < \infty$ which defines a critical threshold: if λ is less than λ_c^{cp}, then λ is described as subcritical, and:

$$P[\tau_A^{cp}(\lambda) < \infty] = 1 \quad (2)$$

Otherwise, if $\lambda > \lambda_c^{cp}$, that is, in the supercritical case:

$$P[\tau_A^{cp}(\lambda) = \infty] > 0 \quad (3)$$

In the first case, the process dies out in a finite time almost surely. In the second case, there is a positive probability that the process never becomes extinct. While the second case does not guarantee the survival, the first case guarantees the extinction: this should be enough motivation to identifying the value which separates both cases.

3. AD HOC DISSEMINATION PROCESS

We model here the ad hoc dissemination process as a variation of the contact process on the 2-dimensional lattice as well as on the n-ary tree. More general topology are not studied yet: these two topologies are the stepping stones to gain insight into the process. We consider the dissemination of a single object. Thus a node state is a boolean variable, with value 1 or 0 depending on the node holding a live copy of the object or not. Note that the results would generalize to the case of multiple independent data objects by constructing a product measure on these states. There is no loss of generality in considering only one single object.

3.1 Ad hoc dissemination and its relation to the contact process

We consider an ad-hoc network arranged according to a 2-dimensional lattice. We also assume that there is no node mobility.

A node v with coordinates (i, j) wishes to use some object periodically. We assume for now one unique object is being replicated throughout the network. The request process for this object is a Poisson process with rate λ. The request takes form of a broadcast message that is being sent to the nodes surrounding v. In order not to flood the whole network with requests, we require that the request message cannot be forwarded to a distance greater than D hops. D is the search depth, or the search radius.

We define the process X_t over the lattice $\mathbf{Z}^2$ by: $X_t(w) = 0$ if at time t, node w is unable to locate the requested object, and $X_t(w) = 1$ if at time t, node w is able to provide a copy of the object. It then returns a reply message to the node v containing the desired object replica.

We define the distance between $v_1 = (i_1, j_1)$ and $v_2 = (i_2, j_2)$ to be:

$$d(v_1, v_2) = |i_2 - i_1| + |j_2 - j_1| \quad (4)$$

and we denote by B_v^r the ball of center v and radius r, namely:

$$B_v^r = \{w \in \mathbf{Z}^2 : d(v, w) \le r\} \quad (5)$$

At the time t of the request, we have either $X_t(v) = 1$, in which case node v knows how to locate the application, or $X_t(v) = 0$.

If $X_t(v) = 0$ and there exists a node w within B_v^D such that $X_t(w) = 1$, then v receives a positive reply to its request, and $X_{t+}(v) = 1$.

If $X_t(w) = 0$ for all the nodes in B_v^D, including v, then there is no reply to the request, and node v is unable to locate the desired object. In this case, the network fails to provide the object to the node, and $X_{t+}(v)$ stays equal to 0.

Finally, we assume that the object expires after some time-out value. This embodies the individual constraint of each node, which cannot store the object indefinitely due to its own limitations. We assume that information times out according to an exponentially distributed random variable with rate 1. There is no loss of generality in choosing the rate equal to 1 as we will see the properties of the network depend only on the ratio of the request rate vs. the time-out rate.

As in the contact process, we can define an extinction time when starting from the initial condition $A = \{v : X_0(v) = 1\}$:

$$\tau_A(\lambda) = \inf\{t : \forall x, y \in \mathbf{Z}^2, X_t(x, y) = 0\} \quad (6)$$

REMARK 3.1. *The extinction time $\tau_A(\lambda)$ is increasing with λ.*

This is due to the fact a Poisson process with rate $\lambda' > \lambda$ is the sum of two Poisson processes with rate λ and $\lambda' - \lambda$. A coupling argument shows that adding the second Poisson process only increases the lifetime of the process.

4. CRITICAL RATE

Consider for now that the rate λ is a varying parameter.

4.1 Lattice topology

In this section, we prove that there exists a critical rate value λ_c. For request rates $\lambda < \lambda_c$, the process becomes extinct (ie. $X_t = 0$) with probability 1 as $t \to \infty$, no matter what is the finite set A of initial conditions. For request rates $\lambda > \lambda_c$, then the probability that $X_t \neq 0$ is strictly positive.

We write this formally: denote by $\tau_A(\lambda) = \min\{t : \forall x \in \mathbf{Z}^2, X_t(x) = 0\}$. τ is the extinction time of the process starting with all nodes in A infected. Since for $X_t(v)$ to transitions from 0 to 1, it needs at least one node w in the D-radius ball surrounding v for which $X_t(w) = 1$, this ensures that: $\forall t > \tau, \forall x \in \mathbf{Z}^2, X_t(x) = 0$.

THEOREM 4.1. *There exists $\lambda_c \in \mathbf{R}, 0 < \lambda_c < \infty$ such that, for a finite set of initial condition A:*

$$\begin{aligned} \text{if } \lambda < \lambda_c, P[\tau_A(\lambda) < \infty] &= 1 \\ \text{if } \lambda > \lambda_c, P[\tau_A(\lambda) = \infty] &= \alpha > 0 \end{aligned} \quad (7)$$

REMARK 4.1. *A necessary condition for the ad hoc dissemination process to survive is thus to define the time-out value such that $\lambda > \lambda_c$. We will still need to study how the distance from a node v to a point x such that $X_t(x) = 1$ evolves with time.*

Proof: We prove this by bounding λ_c above and below.

• First, we find an above bound. We compare the process X with X^{cp} starting from the same initial condition. We note that the ad hoc dissemination process X is monotonic with D the broadcast radius. This means that $X_t^{D'}(v) \geq_{st} X_t^D(v)$ if $D' \geq D$. As a consequence, if λ_c^D exists, then it is decreasing with D. Thus, if λ_c^D exists, it is bounded above by λ_c^1.

We now compare the ad hoc dissemination process with radius 1 with the contact process described above. The main difference is that the contact process is a *push* process, meaning that infected nodes spread the disease to susceptible nodes, while the ad hoc dissemination process is a *pull* process, meaning that the node requests the object from its neighbor. Thus, the object request rate is not dependent on the cardinality of the infected neighbor set.

However, if we set $\lambda = 4\lambda^{cp}$, then the ad hoc dissemination process with radius 1 requests the object from its neighbor always more than the contact process would infect this node. A coupling argument shows that $X_t^{1,4\lambda}(v) \geq_{st} X_t^{cp,\lambda}$. This proves that $\lambda_c^1 \leq 4\lambda_c^{cp}$.

• Secondly, we prove a lower bound on λ. We consider the case where the broadcast radius of the radius is set to D. Consider the contact process with $\lambda^{cp} = B\lambda$, where $B = |B_0^D|$. We need to show that if the ad hoc dissemination process never becomes extinct, neither does the contact process. This will yield $\lambda_c^{cp}/B < \lambda_c^D$.

To see this, we restrict ourselves temporarily to a $N \times N$ lattice instead of $\mathbf{Z}^2$. We consider a sample path for which the ad hoc dissemination process never becomes extinct. Start two processes with the same initial conditions: one is the ad hoc dissemination process, the other we construct as follows (we use prime notations to differentiate the second process from the first): for a node v for which $X = 1$ that replies to an object request, map it to a node v' in the contact process graph which spreads its infection to another node. The mapping is as follows: $v = v'$ at time 0, as the initial condition are identical. Then, for each node w that requests the object from v, pick a neighbor of v' in the direction of w. If more than one directions are equivalent, pick either non-infected node. If all nodes are infected, pick the first non infected node out of the infected cluster in the direction of w. If several nodes fit the description, apply a random tie-breaker. Since we started with the same initial conditions, and since up to this point in time, there is exactly the same number of infected nodes, then we can find a non-infected node in the construction as there is one in the ad hoc dissemination process requesting the object.

This ensures that there are always as many infected nodes in each process. The rate of request that node v replies to is at most $B\lambda$, which is the spreading rate in the contact process. Thus the contact process will have more nodes infected than the process we just constructed. This is true for all N and the inequality holds as $N \to \infty$. Thus, if the ad hoc dissemination process survives, so does the contact process with rate $B\lambda$, and $\lambda_c^{cp} < B\lambda_c^D$.

Since [4] proves that $0 < \lambda_c^{cp} < \infty$, this completes the proof of the Theorem.

∎

4.2 Process on a tree topology

A network topology of interest is the tree topology. A self-configuring ad hoc network might configure itself as a tree: a node v which seeks to connect to the ad hoc network will look for a node that is already connected. When it finds such a connected node w, it becomes a leaf to this node, and the node w becomes a root to the new node v. Once a node has a maximum number of branches, it refuses connection to more nodes, which will find another non-saturated node to connect to the network. The root node of the network is typically a node with connection to the wider internet. In a sensor network, the root would be the data sink, and the tree structure follows from the routing structure to and from the sink.

In some other ad-hoc network, a minimal spanning tree is build to avoid repetition of broadcast messages and to save overhead and power. The broadcast of an object request is then forwarded over the minimal spanning tree, even though the topology of the network might be different, or such tree might not be unique.

We consider a tree of fixed degree. We denote by ρ the root vertex of the tree. The degree of the tree is $n \geq 2$.

THEOREM 4.2. *The proof of theorem 4.1 applies on a tree topology instead of a lattice topology, provided the contact process satisfies equations 2, 3.*

Proof: The proof of 4.1 has to be adapted to the number of neighbors of a node on a tree, which depends on how many edges are connected to each vertex. On a binary tree with $n = 2$, it would be 3, instead of 4 neighbors in the proof of 4.1. The other properties and quantities translate naturally to this setting. ∎

THEOREM 4.3. *The contact process on a tree has a critical probability* $0 < \lambda_c^{cp} < \infty$

Proof: This is proven in [7]. ∎

Theorem 4.2 says that the ad hoc dissemination process will survive for an infinite period of time with some positive probability on the infinite tree of degree n. However, it does not state that it reaches a steady-state: we will prove that for some values of λ, the ad hoc dissemination process drifts down the tree. Only nodes further and further away from the tree root carry the object. The process stays alive, but in a transient state. For each given edge, the distance from the edge to the information then goes to ∞, and an object request broadcast by this edge is unable to find a reply within D hops.

We now state the Theorem:

THEOREM 4.4. *For*

$$\lambda < \frac{1}{\sum_{k=1}^{D}(k+1)n^{\frac{k}{2}}} \tag{8}$$

the ad hoc dissemination process does not converge to a steady state

Proof: The proof is adapted from [7]. We define a weighting on all the vertices by assigning weight $W(v) = n^{-d/2}$ to the vertex v, where d is the distance from the vertex v to the root of the tree ρ.

We define $M(t)$ to be the total weight over the tree.

$$M(t) = \sum_{v} W(v)\mathbf{1}_{\{X_t(v)=1\}} \tag{9}$$

We will prove that M can be made to be a supermartingale for the values of λ satisfying equation 8 and as such, the weight must converge to 0.

$$\begin{aligned} & E[M(t+dt)|X_t] \\ &= E[\sum_{v} \mathbf{1}_{\{X_t(v)=1\}} W(v)|X_t] \\ \leq \sum_{v} \mathbf{1}_{\{X_{t+dt}=1\}} W(v) & \left(-1 + \lambda \sum_{k=1}^{D}(k+1)n^{k/2}\right) \\ = M(t) + dtM(t) & \left(-1 + \lambda \sum_{k=1}^{D}(k+1)n^{k/2}\right) \end{aligned} \tag{10}$$

The step from the second line to the next in the equations above requires an explanation: the possible transitions for the vertex v between t and $t+dt$ are either (i) the timing-out of the object at vertex v, (ii) the request of the object from a vertex less than D hops away for which the shortest path from v goes through a node closer to the origin, and (iii) the request of the object from a vertex less than D hops away from a node for which the shortest path only goes further away from the origin.

•(i) happens at rate 1, and takes away with it the weight $W(v)$. This is accounted by the term -1 in between the parenthesis, with $W(v)$ factored out.

•(ii) will add the weight of the nodes successfully requesting the object upstream from v . Assuming all the other nodes y within D hops have $X_t(y) = 0$ will yield an upper bound on the total weight that $M(t)$ can increase. For each node y such that the path from v goes at least one step closer to ρ, we have an increase of the weight that can be computed by summing on all the possible vertices to be the closest node to ρ on the path from v to y. Denote by $c_\rho(v, y)$ the closest node to ρ on the shortest path between v and y. Figure 2 shows the possible values of $c_\rho(v, y)$ as well as the possible candidates for the y vertices.

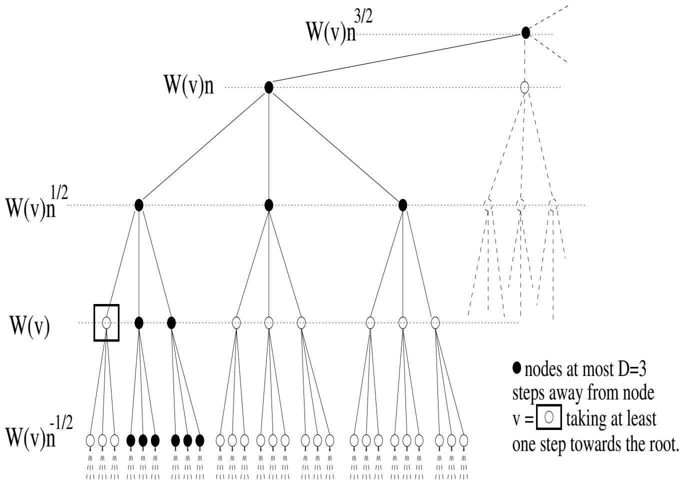

Figure 2: Vertices at most D hops away from v towards ρ

If $c_\rho(v, y) = y$, then a weight $W(y) = W(v)n^{D/2}$ gets added to the total. If $c_\rho(v, y)$ is $D - k$ hops away from v, then it adds the weight from the subtree from $c_\rho(v, y)$ with depth k. This weight is less than:

$$\begin{aligned} \lambda W(v) & \left(n^{(D-k)/2} + nn^{(D-k-1)/2} + \dots \right. \\ & \left. \dots + n^k n^{((D-k)-k)/2}\right) \\ &= \lambda W(v) \left(\sum_{j=0}^{k} n^{(D-j)/2}\right) \end{aligned} \tag{11}$$

Some weight are counted twice when summing on all possible values of $c_\rho(v, y)$, which only makes the upper bound a little looser. This summation yields a weight bounded above by:

$$\lambda W(v) \left(\sum_{j=1}^{D} jn^{j/2}\right) \tag{12}$$

•(iii) The subtree composed uniquely of leaves of node v gives a maximum weight increase of:

$$\lambda W(v)\left(nn^{-1/2}+n^2n^{-2/2}+\ldots+n^Dn^{-D/2}\right)$$
$$=\lambda W(v)\left(\sum_{j=1}^{D}n^{j/2}\right) \quad (13)$$

Combining the term from (i) with equation 12 and 13 yields the last equality in equation 10.

For the value of λ such that

$$\lambda < \frac{1}{\sum_{k=1}^{D}(k+1)n^{k/2}}$$

the term $(-1+\lambda\sum_{k=1}^{D}(k+1)n^{k/2})<0$ and as a consequence, $E[M(t)]$ decreases exponentially fast to 0.

This means that the weight contribution of any fixed vertex is eventually 0. ∎

5. RELATION BETWEEN D AND λ_C

To ensure the survivability of the ad hoc dissemination process, we have allowed the rate λ to take different values. However, in a practical setting, the rate is a fixed constant that is determined by the users' behavior. We will turn our attention to the case where λ is constant.

5.1 Lattice topology

We now see how we can apply this critical phase phenomenon to the optimization of the ad hoc dissemination problem. There are two parameters that are available to us: D the broadcast radius, and the initial condition. λ is the rate at which a node will need the object, and it is intrinsic to the nodes. We study here the impact of D only.

We have seen that λ_c^D is a decreasing function in D. We wish to find out the best broadcast radius given that λ, the request rate for the object, is set.

THEOREM 5.1. *For any λ, and for a finite set A, we can find D such that, for $D'>D$,*

$$P[\tau_A^{D'}(\lambda)<\infty]=\alpha>0 \quad (14)$$

Proof: We define by $S_{k,l}$ the square composed of the nodes of coordinates $kD+i, lD+j$ where $i,j\in\{0,\ldots,D-1\}$. We define the process $Y_t(k,l)$ to be equal to 1 if $X_t(v)=1$ for at least one $v\in S_{k,l}$ and 0 otherwise. The process Y is an aggregation of X over the squares $S_{k,l}$.

We start the process Y at time 0 and define a process Y' that is equivalent to Y with the following difference: if $Y_t(k,l)$ turns from 0 to 1 by receiving the object reply from a node in either $S_{k+1,l+1}, S_{k-1,l+1}, S_{k+1,l-1}, S_{k-1,l-1}$, then Y' stays equal to 0. This means that $Y'_t(k,l)$ can be infected only by its single hop neighbors on the (k,l) lattice, not by the nodes that are diagonally connected to it. $Y'\leq Y$ by construction, for all t and all (k,l).

Now, we also assume that Y' goes from 1 to 0 according to the process of one of the infected node in $S_{k,l}$. Which one does not matter as they are all distributed independently with a Poisson process with the same rate. Again, Y' goes to 0 faster than Y, since Y will turn to 0 only when all the nodes in the square have a value 0 for X (including the one we chose for Y').

The rate of requests in the process Y' from its single hop neighbor is $D^2\lambda$ (we consider only the request that infect the neighbor). Thus Y' is an ad hoc dissemination process with radius 1, extinction rate 1 and rate $D^2\lambda$. We have shown that $\infty>\lambda_c^1>0$ existed, and thus, taking D big enough such that $D^2\lambda>\lambda_c^1$, the process Y' is supercritical.

This in turn implies that Y is supercritical, and X must be as well. ∎

5.2 Tree topology

We now turn our attention to tree topology. We can first state the equivalent of Theorem 5.1 for the tree topology:

THEOREM 5.2. *For any λ we can find D such that, for $D'>D$,*

$$P[\tau_A^{D'}(\lambda)<\infty]=\alpha>0 \quad (15)$$

Proof: We number the vertices on the tree n^k+l, with k the distance to the root ρ and $0\leq l<n^k$. Thus (k,l) uniquely define a vertex in the tree. We define by $T_{k,l}$ the subtree of depth D rooted at the vertex $(kD,l)=n^{kD}+l$, for $k\geq 0, 0\leq l<n^{Dk}$. This is the set of leaves of vertex $v=n^{kD}+l$ within D hops of $n^{kD}+l$. We define the process $Y_t(k,l)$ to be equal to 1 if $X_t(v)=1$ for at least one $v\in T_{k,l}$ and 0 otherwise. The process Y aggregates X over each subset of the tree $T_{k,l}$. The process Y is built upon a tree as well (but one of degree n^D).

As in the proof of Theorem 5.1, we start the process X and Y at time 0 and define a process Y' that is equivalent to Y with the following difference: if $Y_t(k,l)$ turns from 0 to 1 by receiving the object reply from a node in any $T_{k,j}$, with $j\neq l$, then $Y'_t(k,l)$ stays equal to 0. This means that $Y'_t(k,l)$ can receive object replies only from the node in the Y tree either directly upstream from the vertex (k,l) or downstream, but not from the nodes at the same distance from the root. $Y'\leq Y$ for all t and (k,l).

Now, we also assume that Y' goes from 1 to 0 according to the process of one of the infected node in $T_{k,l}$. Which one does not matter as they are all distributed independently with a Poisson process with the same rate. Again, Y' goes to 0 faster than Y, since Y will turn to 0 only when all the nodes in the square have a value 0 for X (including the one we chose for Y').

The rate of request in the process Y' from one of its single hop neighbor is more than $\lambda/2\sum_{k=1}^{D}n^k$ (it is twice as much from a leaf subtree, and this much from the subtree above in the tree). A coupling argument allows us to consider the value $\lambda/2\sum_{k=1}^{D}n^k$ for Y'. Now, Y' is a ad hoc dissemination process with radius 1 and rate $\lambda/2\sum_{k=1}^{D}n^k$. We have shown that $\infty>\lambda_c^1>0$ existed for the tree of degree n^D [1]. Thus, taking D such that $\lambda/2\sum_{k=1}^{D}n^k>\lambda_c^1$ ensures that the process Y' is supercritical. This in turns imply that Y and X are as well.

This implies that there is a minimal broadcast radius for which the network operates in the supercritical mode. Since the broadcast of a request is a burden on the network in terms of bandwidth usage, power usage if the nodes are wireless, processing power usage, the minimal broadcast value should be identified as the preferred mode of operation.

The value of D cannot be computed from Theorem 5.2. We now state a necessary condition for D for the process to

[1] λ_c^1 is decreasing with the degree of the tree, thus we can take the value for the tree of degree n to avoid the dependence on D

survive indefinitely. It is easy to compute the value of D for this necessary condition:

THEOREM 5.3. *On a tree of degree n, in order to have, for some $\alpha > 0$*

$$P[X_t(v) = 1] > \alpha, \forall v \text{ as } t \to \infty \quad (16)$$

it is necessary to pick D such that

$$\sum_{k=1}^{D} (k+1)n^{k/2} > \frac{1}{\lambda} \quad (17)$$

Proof: This is a trivial consequence of Theorem 4.4. ■

Figure 3 plots the function $f_D(n) = 1/\sum_{k=1}^{D}(k+1)n^{k/2}$ for n equal to 1 to 5, and D varying between 1 and 10. It shows the minimal value for λ to satisfy the necessary condition in Theorem 5.3. For a given value of λ, and for a given degree n, the minimal search depth (ie., the minimal broadcast radius) is found by finding the first value D such that the point $f_D(n) < \lambda$.

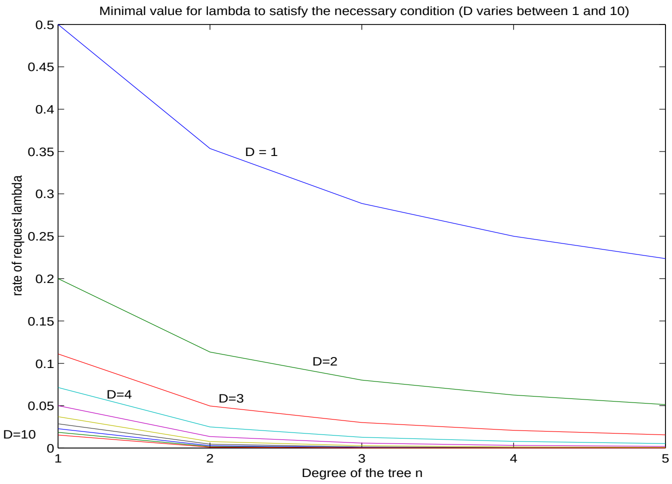

Figure 3: Necessary condition for D, n and λ

6. NUMERICAL RESULTS

To assess the relation of the depth D with the viability of the process, we ran some simulations both on the lattice topology and on the tree. In both topologies, the simulation was conducted over a network of 10,000 nodes and were run until a steady state was reached. Simulations were run several times for each set of values D and λ and for each initial conditions. The initial condition was chosen so that a large fraction of the nodes were infected (ie. possess the object).

In figure 4, we ran the ad hoc dissemination process over the lattice. The initial condition was chosen so that one node every 8 nodes was infected. The rate λ is set to 0.33. We see that for $D > 2$ the ad hoc dissemination process reaches a steady-state and dies out for $D = 1$.

In figure 5, we ran the ad hoc dissemination process on the tree. The tree topology had degree $n = 2$. We see that for the search depth $D = 2$ to 5, the process dies out, and it converges to a steady state for $D \geq 6$.

We run simulations to assess the value of λ_c. We plot on figure 6 the number of infected nodes to which the ad hoc

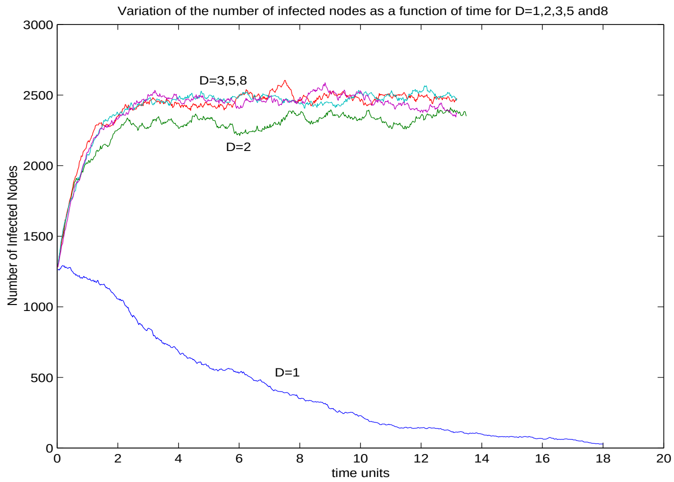

Figure 4: The time behavior of the ad hoc dissemination process on the lattice

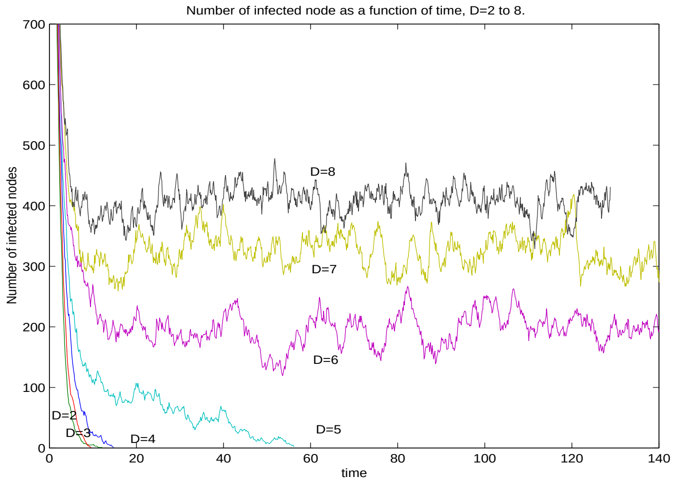

Figure 5: The time behavior of the ad hoc dissemination process on the binary tree

dissemination process converges as a function of λ for the fixed search depth $D = 4$ on the lattice. We estimate the value of $\lambda_c^{D=4}$ to be about 0.04.

If we now go back all the way to figure 1, it shows the distribution of the information after running the ad hoc dissemination process long enough so it converges, with $\lambda = 0.33$ and $D = 3$. The number of nodes infected is a quarter of all nodes. In the lattice, the drift that we discussed about the tree topology, cannot happen due to the symmetry of center the origin.

These numerical evaluations should be refined for the particular properties of the network and of the object.

7. CONCLUSIONS

We first motivated the problem of distributed storage of information in networks of nodes with individual constraints. We have presented a stochastic modelling for the ad hoc dissemination process and identified the existence of a critical request rate for such process: for object requests that are of

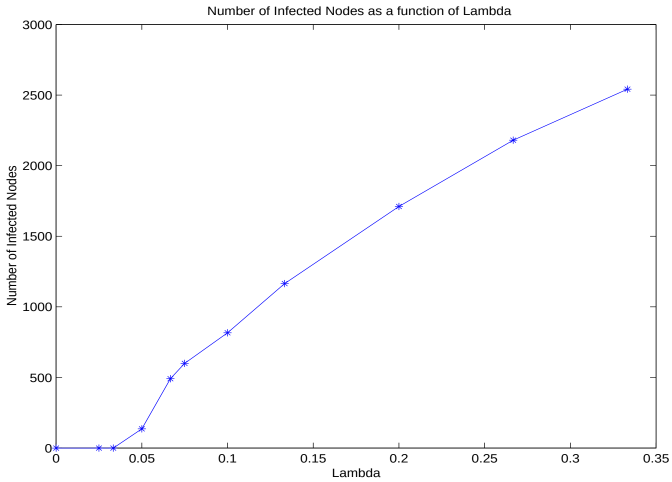

Figure 6: Number of nodes infected as a function of λ

rate less than the critical value corresponding to the search radius, the ad hoc dissemination process almost surely dies in finite time, and the information stored by the process disappears.

We also investigated the relationship between the search radius D and the request rate λ. We showed that there is a critical search radius for any positive rate λ such that, for all radii smaller than the critical search radius, the ad hoc dissemination process dies out almost surely. We computed some numerical bounds on the critical radius for the tree topology. Extending the numerical bounds for other topology, and tightening up the bounds are the next steps in our research agenda.

Another issue is to consider the minimal cost, in messages sent, to maintain the object alive in the network. We have shown in this paper that this cost is at most constant per node. This is by the definition of our request mechanism. The interesting question is to allow different nodes to have different broadcast radius, and to find the minimal cost to the network that maintain the object alive.

We did not consider node mobility in this document: this is a direction that we would like to turn our attention to. Of great interest is how the mobility relates to the peremption of the information. For instance, if the information we look for is routing information, then the node movement will impact the validity of the information: what should be the rate of connection from one node to another that ensures that the routing information stored in intermediary routers is accurate? We are confident that our model can be adapted to answer such questions.

With mobility comes the issue of the network topology: in this first look at the problem, we chose two basic topologies. We will address nodes distributed as a 2-dimensional Poisson process with a fixed connectivity radius (as described in [2]) in a paper currently in progress. Also of interest would be to study a tree with a random degree for each node, as a Galton-Watson tree. We believe our technique could apply there with the appropriate modifications.

8. REFERENCES

[1] Chen B., Jamieson K., Balakrishnan H., Morris R. *Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks*, ACM Wireless Networks Journal, Volume 8, Number 5, September, 2002. Pages 481 - 494.

[2] Gilbert E. *Random Plane Networks* J. Soc. Industrial Applied Mathematics, Vol. 9, No. 4, 1961, pp.533-43.

[3] Handjani S. *Survival of Threshold Contact Processes* Journal of Theoretical Probability 10 (3): 737-746, July 1997.

[4] Harris T.E. *Contact Interactions on a Lattice* in Annals of Probability, 1974, Vol.2, No. 6, 969-988.

[5] Koodli R., Perkins C. *Service Discovery in On-Demand Ad Hoc Networks* IETF Manet Working Group, Internet Draft draft-koodli-manet-servicediscovery-00.txt, 9/2002.

[6] Liggett T. *Stochastic Interacting Systems* Springer, 1999.

[7] Pemantle R. *The Contact Process on Trees* Ann. Probab., 20, 2089 - 2116 (1992)

[8] Perkins C., Royer E. *Ad hoc On-Demand Distance Vector Routing.* Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, February 1999, pp. 90-100.

[9] Plaxton C., Rajaraman R., Richa A. *Accessing Nearby Copies of Replicated Objects in a Distributed Environment* ACM Symposium on Parallel Algorithms and Architectures, 1997.

[10] Stoica I., Morris R., Karger D., Kaashoek F., Balakrishnan H. *Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications* in Proceedings of ACM SIGCOMM, 2001.

[11] Wagner A., Anantharam V. *Designing a Contact Process : The Piecewise-Homogeneous Process on a Finite Set with Applications.* Stochastic Processes and their Applications (to appear, available from http://www.eecs.berkeley.edu/ ananth/).