Epidemic Models, Algorithms and Protocols in Wireless Sensor and Ad-hoc Networks

Pradip De and Sajal K. Das
Center for Research in Wireless Mobility and Networking(CReWMaN)
Department of Computer Science and Engineering
University of Texas at Arlington, TX 76019-0015
{pradipde, das}@cse.uta.edu *

1 Introduction

Sensor networks are composed of a large number of sensing devices, which are equipped with limited computing and radio communication capabilities. They have diverse application areas, ranging from tracking and intrusion detection for security purposes to environment monitoring and traffic and location systems. However, with the steady advancements in processor, memory, communication and sensing technology, and a drive towards a smarter environment, there is an increased interest in the development and deployment of wireless sensor networks to be used for many interesting and new applications. These applications range from real-time remote monitoring and control, military surveillance, environmental monitoring to healthcare management, construction safety, etc.

Within the next few years, it is very likely that the number of deployed sensors will see an exponential increase. Most of these networks will require application-specific functionalities and performance requirements [2]. However, the realization of these networks poses a lot of challenges in system and network design, algorithm and protocol design, and query language and database design. The primary issues under focus which are critical to the proper functioning of wireless sensor networks are energy consumption, connectivity, clustering techniques, data aggregation, etc. These issues stem mostly from the stringent resource constraints of the sensor nodes. Therefore, in order to address these issues, we require efficient modeling techniques and robust algorithms and protocols before actual implementation and deployment is done.

Often times, in the course of modeling complex entities and networks, we have taken recourse to biologically inspired paradigms. Biologically inspired modeling techniques are among the many mechanisms that have been adopted to accurately model certain phenomena in wireless sensor networks. For example, data dissemination, routing algorithms, broadcast protocols are among the few areas that have been effectively modeled by epidemic theoretic concepts.

In this chapter, we address the modeling techniques, algorithms and protocols proposed in wireless sensor and ad-hoc networks that are primarily based on Epidemic Theoretic concepts and paradigms.

The rest of the chapter is organized as follows. In Section 2, we provide a general overview of Epidemic Theoretic concepts and analysis. In Section 3, we discuss the data dissemination models in sensor networks and their use of epidemic theory. Section 4 illustrates several reprogramming and code update protocols in sensor networks that adopt epidemic theoretic principles. In Section 5, we look into epidemic protocols in ad hoc networks. Section 6 looks into some security aspects and explains the propagation process modeling of malware in sensor networks. Finally, we conclude the chapter in Section 7.

*This work is supported by NSF ITR grant No. IIS-0326505 and Texas ARP grant No. 14-748779

2 Overview of Epidemic Theory

In order to appreciate the epidemiological models applied in wireless sensor networks, we need to first understand the concept of epidemiology. In this section we provide a terse description of the theory and its applications. Epidemic Theory[1] is the study of the dynamics of how contagious diseases spread in a population, resulting in an epidemic. Primarily, the theory mathematically models the propagation process of an infection and measures its outcome in relation to a *population at risk*. The population at risk basically comprises of the set of people who possess a susceptibility factor with respect to the infection. This factor is dependent on several parameters like exposure, spreading rate, previous frequency of occurrence etc., which define the potential of the disease causing the infection. Among the different models characterizing the infection spread, two are quite popular. They are the *Susceptible Infected Susceptible* (S-I-S) Model, *Susceptible Infected Recovered* (S-I-R) Model etc. In the former, a susceptible individual acquires infection and then after an infectious period, (i.e., the time the infection persists), the individual becomes susceptible again. On the other hand, in the latter, the individual recovers and becomes immune to further infections.

An approach to model the propagation of an infection is to assume that the probability (per unit time) for a susceptible individual to acquire infection is equal to the average rate at which new infective partners are acquired multiplied by the probability of being infected by any one such partner. In the general deterministic S-I-R model, if *N(t)*, *X(t)*, *Y(t)* and *Z(t)* denote the total population, the susceptibles, the infected and the recovered or immune individuals, respectively at time t, we can say

$$N(t) = X(t) + Y(t) + Z(t) \tag{1}$$

If β denotes the infection rate and γ denotes the removal rate of infected individuals, then assuming a homogeneous mixing model i.e., each of the susceptibles can get in contact with any of the infectives, it is simple to observe that in time Δt, there are $\beta xy\Delta t$ new infections and $\gamma y\Delta t$ removals. Therefore, the basic differential equations that describe the rate of change of susceptibles, infectives and recovered individuals are given by:

$$\begin{aligned} \frac{dX(t)}{dt} &= -\beta XY \\ \frac{dY(t)}{dt} &= \beta XY - \gamma Y \\ \frac{dZ(t)}{dt} &= \gamma Y \qquad . \end{aligned}$$

The above equations can be solved either approximately or precisely based on some boundary conditions, such as, at the start of the epidemic, when $t = 0$, (X, Y, Z) can take the values $(x_0, y_0, 0)$. Note that, in particular if y_0 is very small, x_0 is approximately equal to N. It also follows that if the *relative removal rate*, $\mu = \gamma/\beta$, is greater than x_0 only then an epidemic can start to build up as this condition will result in $[dY(t)/dt]_{t=0} > 0$, i.e. $Y(t)$ will have a positive slope. Therefore, the relative removal rate $\mu = x_0$ gives a threshold density of susceptibles.

On the other hand, the S-I-S model does not have the recovered subset $Z(t)$ and those who are infected fall back into the susceptible subset $S(t)$ after their infectivity duration.

An important aspect which is of particular interest in epidemiological studies is the phenomenon of phase transition of the spreading process that is dependent on a threshold value of the epidemic parameter, i.e., if the epidemic parameter is above the threshold, the infection will spread out and become persistent; on the contrary, if the parameter is below the threshold, the infection will die out. Identification of this threshold value is critical in the study of how an epidemic spreads and how it can be controlled.

Apart from the continuous differential rate equation based modeling technique, the study of epidemics has often been performed by treating the population as a network graph, with the nodes representing each individual and the edges their interaction. This form of analysis [9] has mainly been used in scenarios where the end result of the

epidemic spread is more important than the temporal dynamics of the propagation. Several works have spawned from this formulation[6], [5], [7], [8], [10], [11] where the spread of diseases have been studied by modeling the social network as a scale free topology. Several other works also exist that model the spread of computer viruses [12], [31].

Epidemic Theory has found special attention in the design and modeling of several phenomena and protocols in sensor networks wherever there is a scope of information distribution in a large scale preferably from a small number of sources to a large number of recipients. Among the popular phenomena in sensor and ad-hoc networks where this theory has been adopted are data dissemination, broadcast protocols and routing. We will delve into some of these areas where Epidemic Theory has been used to study and model several processes and functions of sensor networks.

3 Data Dissemination in Sensor Networks : Model and Protocols

The problem of reliable data dissemination in the context of wireless sensor networks is very critical. Reliable data dissemination to all nodes is absolutely necessary for the propagation of queries, code updates and other sensitive information in a wireless sensor network. This is not a trivial task since the number of nodes in a sensor network can be quite huge and the environment is dynamic, i.e., nodes can die or move, thus making the topology change constantly.

Since data dissemination primarily deals with the transfer of messages from one node to all nodes of a network, algorithms based on epidemiological formulations are a perfect fit. Accordingly, these algorithms have been successfully used in disseminating information in sensor networks and depending on the application, the dissemination can start at a single node, such as a base station, or at multiple sensor nodes. The decentralized and distributed nature of wireless sensor networks fits the context of epidemic algorithms aptly.

One of the prominent works of data dissemination in sensor networks is SPIN [24]. An obvious problem with normal epidemic broadcast based dissemination is the inefficient use of bandwidth and other resources. Therefore, the basic epidemic strategy needs to be optimized for sensor networks. In [24], the authors proposed the concept of meta data or data descriptors to eliminate the chance of redundant transmissions in sensor networks. Their work focusses on the efficient dissemination of individual sensor observations to all the sensors in a network. Their main contribution was based on the basic deficiencies of classic flooding, viz., *Implosion*, *Overlap*, and *Resource Blindness*. Implosion is sending data redundantly to one's neighbors regardless of whether they already received it. Coverage overlap of nodes can make them gather the same data and flood it to common neighbors. Classic flooding can be blind to the availability of resources when it is flooding data across the network.

The use of metadata allows nodes to negotiate between themselves and prevent redundantly transmitting the same information. Also, in SPIN, each node has a local resource manager that keeps track of its resources and helps a node decide whether to transmit or process data. SPIN first broadcasts metadata to its neighbors. Then, if it receives a request for the data from any neighbor it sends the data to that node.

There are four protocols in the SPIN family. The first two, *SPIN-PP* and *SPIN-BC*, tackle the basic problem of data dissemination under ideal conditions. The other two, *SPIN-EC* and *SPIN-RL* are modified versions of the first two. *SPIN-PP* is optimized for communicating in a point-to-point mode, where for each data transmission between neighbors, a three stage handshaking(ADV-REQ-DATA) is performed. As illustrated in Fig. 1, a node sends an *ADV* message whenever it has new data to advertise. Upon receiving an *ADV* message, the neighboring node verifies whether it has already received or requested the advertised data. If not, it responds by sending a *REQ* message for the missing data back to the sender. The initiator of the protocol responds to the *REQ* message with a *DATA* message containing the missing data.

Although this protocol has been designed for a lossless environment, it can be adapted for a lossy environment. Nodes can periodically send the *ADV* message to counter lost *ADV* messages. For lost *REQ* and *DATA* messages, nodes can request for items that do not arrive within a fixed time period. *SPIN-EC* is a modification of *SPIN-PP* so

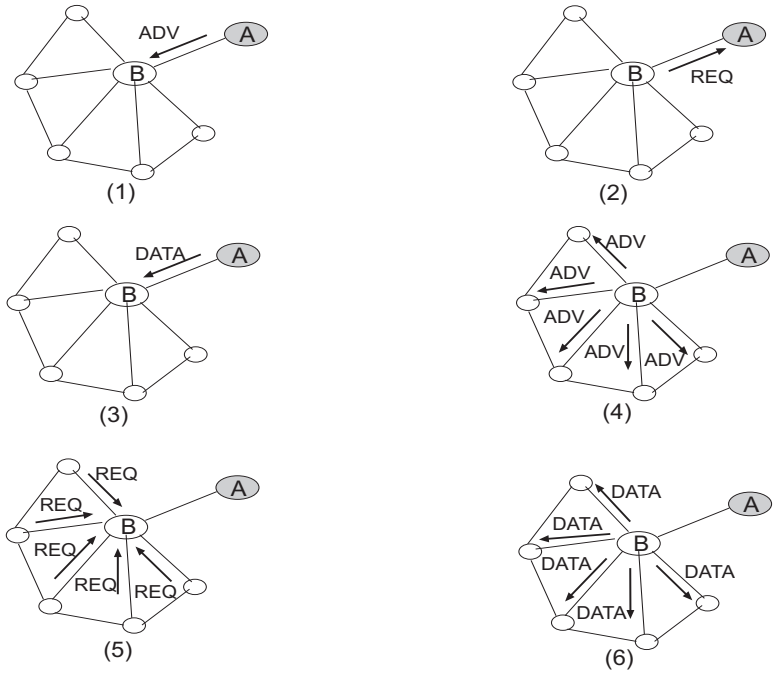

Figure 1. SPIN-PP Protocol. Node A sends Advertisement messages (*ADV*) to B. B responds with a Request (*REQ*) message. Then B starts to send *ADV* to its neighbors.

that when a node observes that it is approaching a low-energy threshold, it reduces its participation in the protocol.

In *SPIN-BC*, which is a broadcast transmission protocol, each node transmits to the broadcast address. Every node that is in the transmission range of the sender processes the received message. This approach is justified because broadcast and unicast transmissions use the same amount of network resources in a broadcast network. The proliferation of redundant messages in the network can be curtailed by *SPIN-BC* because a node A suppresses its own transmission whenever it observes that another node B has transmitted the same message that A itself was supposed to transmit.

We observe the epidemic nature of the dissemination of data in *SPIN*, especially in *SPIN-BC*. Using the three way handshake, a node which has the missing data passes it on to a neighbor which does not have it, thereby infecting it in the process. The working of the three way handshaking protocol basically constitutes the contact and infecting process of the *SPIN* protocol.

3.1 Infuse

For the reliable dissemination of data in sensor networks, the authors of *Infuse* [32] proposed a TDMA based data dissemination protocol for sensor networks. The primary purpose of the protocol was similar to *Deluge* [25], i.e., reliable dissemination of bulk data in a sensor network. We discuss Deluge later in this chapter. In Infuse, the data dissemination protocol is based on a TDMA based medium access layer. Since TDMA ensures a deterministic slot when a sensor node should transmit its packet, it offers a degree of reliability which is used by the data dissemination strategy adopted in Infuse. The authors tackle the problem of random message losses in the presence of channel errors by considering recovery algorithms based on sliding window protocols, modified to use implicit acknowledgments.

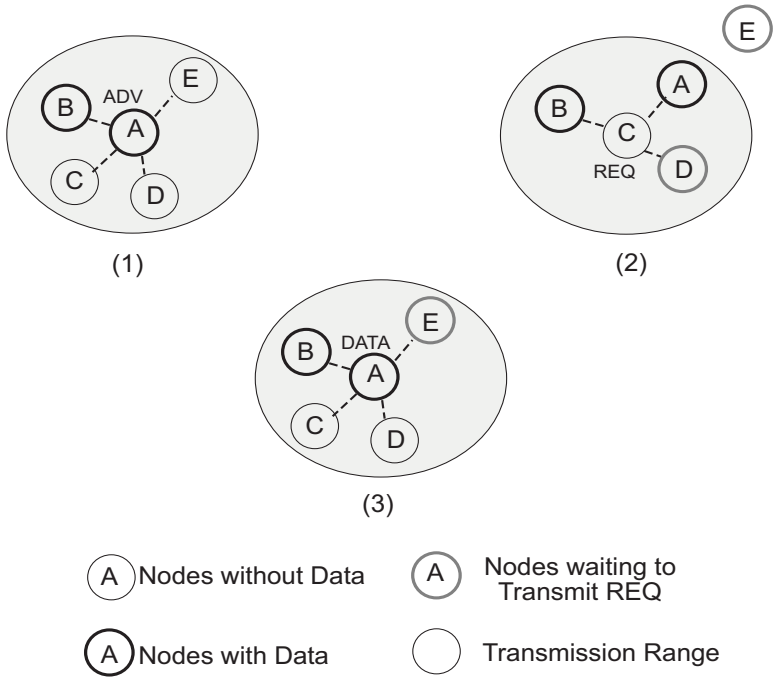

Figure 2. SPIN-BC Protocol.(1) A sends *ADV* to all its neighbors. (2) C responds by broadcasting a request, specifying as originator of the *ADV*. It also suppresses D's *REQ*. (3)After receiving the requested data, E's *REQ* is also suppressed.

In the ideal scenario without channel errors, the base station sends a special *Start Download* message to the sensor which contains the number of subsequent packets to follow in each TDMA slot. The sensor then reserves the necessary flash and downloads the arriving packets.

For dealing with channel errors, Infuse uses an implicit acknowledgment technique. This happens because whenever a successor sensor forwards a data packet, the predecessor node gets to hear it. This overhearing acts as an implicit acknowledgment for the predecessor node. Furthermore, Infuse forwards a received packet in the next TDMA slot thus maintaining a pipeline effect of the transfer process which helps in reducing the total latency of the dissemination process. The use of TDMA based data dissemination also allows Infuse to send the node to sleep except in its own transmission slot, thereby making the Infuse protocol energy efficient.

3.2 Firecracker

Routing a packet from one source to a single destination is fast because forwarding nodes can retransmit without worrying about suppression or local density. At the same time, routing cannot be used to disseminate data to all the nodes in a sensor network because the nodes are not individually addressable. The Firecracker Protocol [22] uses a combination of routing and broadcast principles to rapidly disseminate data throughout a sensor network. As depicted in Fig. 3, a data source first routes the data to be distributed to distant points in the network. Once the data reaches its destination, broadcast based dissemination starts along the path like a string of firecrackers. Firecracker is largely designed to disseminate small pieces of data that would propagate fast, like small programs or configuration constants. While maintaining the energy efficiency of broadcasts, Firecracker can achieve dissemination rates close to routing.

From an epidemic modeling standpoint, Firecracker is fundamentally an infection propagation strategy with a predetermined set of infective nodes defined by the destination nodes of the routing protocol. Having strategically placed the infective nodes at different points of the population, the protocol starts its final broadcast to disseminate the information to the rest of the network. The dissemination strategy could be Trickle [21], and the routing strategy could be any suitable one used for sensor networks.

To elucidate further, Firecracker is composed of three main parts: a) Broadcast Protocol, b) Routing Protocol, and c) Seed Selection. The broadcast protocol is very important to the functioning of Firecracker. It not only

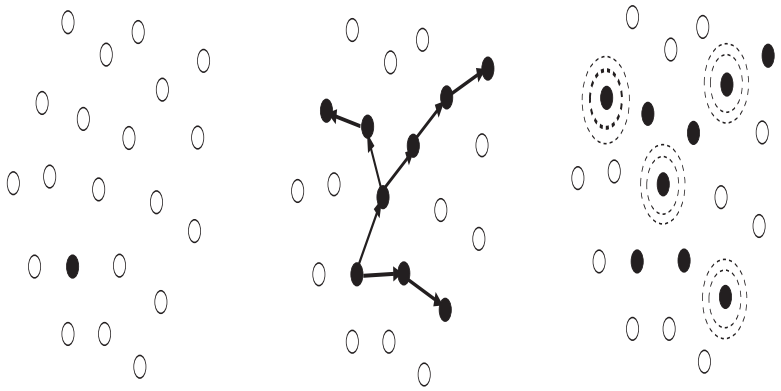

Figure 3. The Firecracker Disseminaton Mechanism

must propagate data to nodes that do not have the data, but also decide when to propagate. Moreover, the protocol should minimize the cost of detection but propagate rapidly and temporary network disconnections should not prevent reception.

The basic purpose of the routing phase is to spread data to distant points in the network so that the initial seeds are placed as deep as possible into the network. This facilitates the following broadcast process to spread the data rapidly. In this regard, a naming scheme that allows nodes to choose such points is helpful. Since wireless data, even during routing phase is essentially broadcast in the neighborhood, nodes along the route should be able to snoop on routed traffic to cache the data as it passes by. Moreover, reliability and non-redundancy are more important than minimum hop paths. Therefore, taking a long, winding path through different areas of the network could benefit the subsequent broadcast protocol in quickly installing the code in all the nodes.

The choice of the seed nodes is also equally important to the performance of Firecracker. The farther the seeds are from the original source and the farther they are spread apart from themselves, the faster the data would propagate to all the nodes of the network.

In general, epidemic algorithms for data dissemination follow the model of nature to spread information and define simple rules for information to flow between nodes of a network. The authors in [17] have done a comparative study of epidemic algorithms for data dissemination. Based on the style of communication between neighboring nodes, they have classified epidemic algorithms for data dissemination into three categories.

- *Pull Based*: A node tries to extract new information from its neighbor.
- *Push Based*: A node sends new information to a selected neighbor.
- *Pull-Push Based*: A node asks its neighbors for new information as well as sends new information to its neighbors.

They have studied the performance of these three classes of epidemic algorithms on sensor networks. Their results show that both pull based and push based algorithms perform better than the push-pull based epidemic

algorithms in terms of delivery rate and scalability. The primary reason for this result is the restricted memory resource of sensor devices.

In [23], the authors performed an experimental and empirical study of the epidemic style algorithms in large scale multihop wireless networks.

A smart tag based data dissemination technique is explained by the authors in [27], where mobile individuals, equipped with smart tags disseminate data across disconnected static nodes spread across a wide area. When the mobile individuals equipped with smart tags move into a sensor field they get updated with the latest information from the sensors. Later, when they move into another field, they disseminate the newly acquired information. The concept of using carriers, who are mobile, to carry data between connected components of the network has also been used by the authors of the epidemic routing protocol[4]. However, they did it more for the purpose of routing whereas here the authors use smart tags to carry the sensed information to another set of output devices like display units. The authors used Bluetooth-enabled smart tags to illustrate the characteristics of their approach. As intuitive, their approach is suited for applications which are delay tolerant.

4 Code Update Protocols in Sensor Networks

Several protocols have been proposed for code update and propagation in sensor networks. These protocols are mainly broadcast in nature, and tasks in sensor networks are assigned through code updates, and all the nodes in the sensor network will have the same code to execute. The propagation mechanism for the code update is basically hop by hop to all nodes in the network. Needless to say, wireless sensor nodes have limited energy, and therefore maintenance costs of the code updates must be low. Another important requirement is rapid propagation of updates, because some tasks may have to be activated as soon as possible and newly assigned tasks make the older ones obsolete. Moreover, the update process should be scalable and should work in a dynamically changing environment.

Being inherently broadcast in nature, these protocols and algorithms fundamentally transmit code updates in a manner similar to an infection spread in a susceptible population. In this subsection, we study some of these protocols and their mechanism.

4.1 Trickle

Trickle [21] is a broadcast algorithm for propagating and maintaining code updates in a wireless sensor network. Conceptually Trickle borrows from epidemiological concepts and performs what the authors claim as *polite gossip*.

Sensor networks are generally deployed in remote areas and are expected to operate unattended for lengthy periods of time. Thus, there is every possibility that the requirements and environments of a sensor network evolve. As a result, users need to be able to introduce new code to retask the network. However, the large scale and embedded nature of the network requires these code updates to propagate through the network. However, as is obvious, networking in sensor networks is very costly in terms of energy consumption and therefore, an efficient and effective reprogramming protocol is necessary.

An effective reprogramming protocol must transfer the code as fast as possible because in the transition time when the code is propagating, the network is actually in a useless state because the old and the new programs are concurrently running.

Propagation of code is costly and nodes need to learn when they need to propagate code. Nodes, therefore, periodically communicate to learn when there is new code. To reduce energy costs, nodes transmit metadata to determine when code is needed. However, the cost of periodically transmitting metadata consumes almost the same amount of energy as actually transmitting the code itself. Therefore, there is a crucial need for the reprogramming algorithm to be efficient in this aspect and effectively determine when nodes should propagate code. Motivated by

this requirement, the authors in [21] have identified three main properties that a reprogramming algorithm should have. They are,

- *Low Maintenance* : When a network is in a stable state, metadata exchanges should be infrequent, just enough to ensure that the network has a single program.
- *Rapid Propagation* : When the network discovers nodes that need update, it should propagate the code as fast as possible and to every node of the network.
- *Scalability* : The algorithm should obviously be scalable and be robust against any environmental changes and node failures.

Trickle tries to meet all these requirements. Its basic working principle is simple. Every so often, a mote transmits code metadata if it has not heard a few other motes transmit the same information. Trickle sends all messages to the local broadcast address. When a neighbor receives a broadcast, either it is up to date, or it detects the need for an update. Detection can be the result of either an out-of-date mote hearing someone having a new code, or an updated mote hearing someone has old code. As long as every mote communicates somehow, the need for an update is always detected. It does not matter who transmits first, but as long as some nodes communicate with each other at a non-zero rate, every node would be up to date. More formally, each node maintains a counter c, a

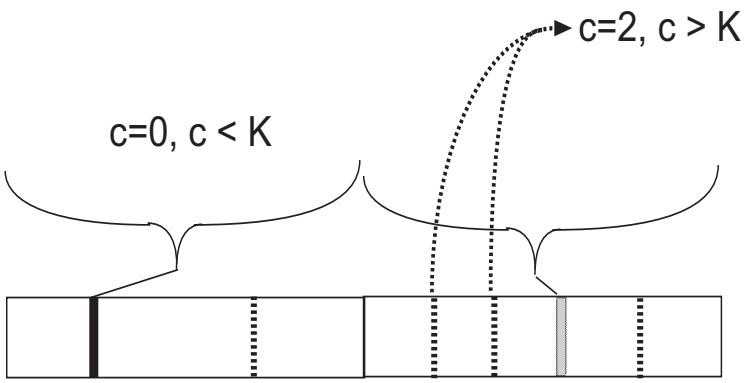

Figure 4. Trickle Metadata advertisement

threshold k, and a timer t in the range of $[0, \tau]$. k is a small, fixed integer (e.g., 1 or 2) and τ is a time constant. When a node hears metadata identical to its own, it increments the counter c. At the timepoint t, which is uniformly randomly chosen in the range of $[0, \tau]$, the mote broadcasts its metadata only if $c < k$. When the interval of size τ completes, c is reset to zero and t is reset to a new random value in the range $[0, \tau]$. Thus, Trickle allows each node to broadcast its metadata at most once per period τ, thus maintaining the politeness of its gossip. In each interval τ , the sum of receptions and sends of each mote is k. The random selection of t uniformly distributes the choice of who broadcasts in a given interval. This evenly spreads the transmission energy load across the network.

In Fig. 4, the solid line represents a transmission, the broken lines represent reception, and the grey line means suppression of an advertisement. This mechanism of Trickle not only allows to scale to high network density, but also propagate updates fast. It also distributes transmission load evenly as it spreads, and is simultaneously robust to transient disconnections. The experimental verification by the authors show that it imposes a maintenance overhead on the order of only a few packets per hour per node.

The epidemiological essence in the working principle of Trickle is evident. The objective is to propagate code as fast as possible to all nodes of the network. Thus, from an epidemic theoretic standpoint, the rate at which the metadata is exchanged gives the rate at which the infection or propagation proceeds in the network. Since after

a node advertises metadata every node in the neighborhood gets updated with the current code, Trickle succeeds in propagating the code update to all nodes in the network. The propagation rate is dependent on the value of τ. With a large value of τ, there is less communication overhead, but the code propagates slowly and conversely in the case of a small τ.

4.2 Deluge

Another type of data dissemination protocol for supporting network programming in sensor networks is Deluge [25]. It is a reliable data dissemination protocol for propagating large data objects from a few source nodes to many other nodes in a wireless sensor network. Trickle's key contribution is its polite gossip that uses suppression and dynamic adjustment of the broadcast rate to limit transmissions among neighboring nodes. It only provides a mechanism for a node to decide when to propagate code. Deluge on the other hand, though based on Trickle's principles, has the added feature of supporting the transfer of large data objects. It uses a three-phase protocol similar to SPIN-RL [24].

Deluge, being an epidemic protocol, can disseminate large data objects as quickly and reliably as possible. The basic local broadcast principle of Deluge is simple and similar to Trickle, but it also addresses several subtle issues that improve its performance. The local suppression of redundant broadcasts makes it density aware. Its three way handshaking mechanism ensures that there is a bidirectional link, thereby making it a reliable data dissemination protocol. Moreover, by dynamically adjusting the rate of advertisements and emphasizing on the use of spatial multiplexing to allow parallel transfers, Deluge allows quick propagation of large blocks of data.

Deluge divides the large data object into fixed size pages for transfer. This enables efficient incremental update and also limits the amount of state that should be reserved at a time at the receiver. Each page is also divided into a fixed number of packets. Because of the epidemic nature of the page propagation, Deluge offers CRC checks at both the packet and page level to be safe from the negative effects of the epidemic nature of data transfer. The

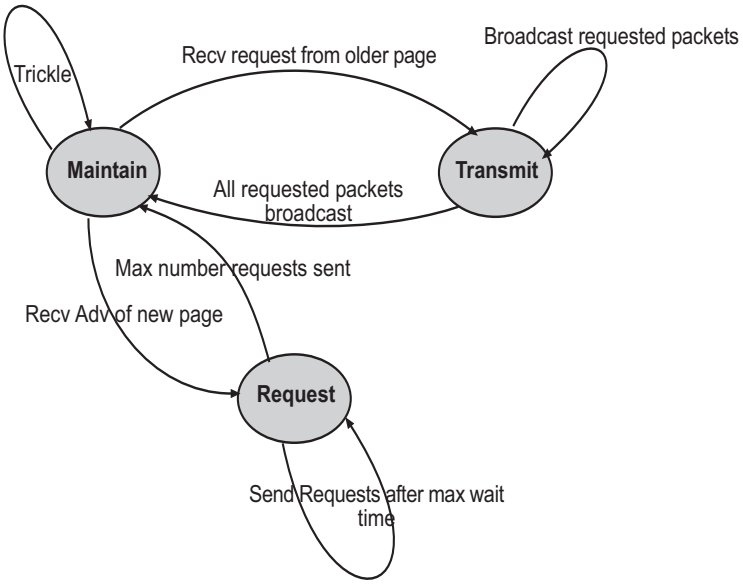

Figure 5. The Deluge State Machine

protocol resides in one of the three states, namely, *MAINTAIN*, *TX*, and *RX*. In the *MAINTAIN* state, a node uses a summary advertisement mechanism to ensure that all nodes in communication range is up to date with the current

version of the object. In the *RX* state, the node is responsible for requesting all remaining packets of a page and while in the *TX* state, it is responsible for broadcasting all requested packets for a given page.

Another work, which was based on an epidemic style multihop reprogramming service for sensor networks was proposed in MNP [33]. One of the basic problems in reprogramming and code update in a wireless network is the issue of message collision and the hidden terminal problem. The authors counter this problem by proposing a sender selection algorithm whereby it is guaranteed that in a neighborhood, there is at most one sensor transmitting at a time. In the basic version of the sender selection protocol, a node becomes a source node and starts advertising this fact only when it acquires the new program code entirely.

Each source node maintains a variable that indicates the number of distinct requests it has received so far and it gets incremented each time a node receives a new download request. Two messages are used for sender selection, namely *advertisement* and *download request*. The *advertisement* message contains information about the new program and the source node. When a node j receives the advertisement request from node i and it is in need of the new code, it sends a *download request* message to the broadcast address so that any neighboring node k becomes aware that i is a potential source.

In order to ensure that a node is aware of all the requesters who are likely to receive the code, if it is chosen to transmit the code, the node sends a download request to all senders of the advertisement messages. However, if node j loses to node k that has more requesters, then whenever j attempts to advertise again, j must reset its *request counter* value to zero, and recalculate its requesters. After k finishes transmitting, it sleeps for a while, so that other sources get a better chance to send. When it wakes up and re-enters the advertising state, its counter value is reset to zero, and a new round of sender selection begins.

We observe that there is considerable similarity with MNP and the SPIN family of protocols because both of them use a kind of three way handshaking procedure for disseminating the data.

For the transfer of large sized data, MNP tries to incorporate pipelining into the data dissemination process by breaking the large data into *segments*, each containing a constant number of packets. Thus, the protocol now operates at the segment level which helps in a node forwarding segments even if it has not received the whole data. In this aspect it is very similar to Deluge [25].

MNP is equipped to address reliability issues like loss detection and recovery. Each packet has a unique ID and each receiver is responsible for detecting its own loss. Since the size of a segment is considerably small, a bitmap of the current segment is maintained in memory, where each bit corresponds to a packet. Using this, a sensor node can receive packets in any order. This bitmap is called the *Missing Vector*. A node also maintains a *Forward Vector* which is a bitmap of the advertised segment. Whenever a node sends a *download request*, it puts its *Missing Vector* in the request message. The advertising node marks its *Forward Vector* according to the *Missing Vector* messages it receives. A node only sends the packets indicated in the *Forward Vector*. Upon receiving all the segments of a program, the node reboots.

5 Epidemic Models in Ad Hoc Networks

In ad hoc networks, the power supply of individual nodes, wireless bandwidth are limited, and the channel conditions can vary significantly. Moreover, since nodes can be mobile, routes may constantly change. Thus, to enable efficient communication, robust routing protocols must be developed. Several existing Mobile ad hoc routing protocols [18], [19] have been developed that allow wireless nodes to communicate with one another without any pre-existing network infrastructure. In this section we look into some of the routing protocols that essentially have the flavor of epidemiology.

5.1 Gossip

Although flooding has been used with some optimization to route packets in an ad hoc network, many routing messages are propagated unnecessarily. The authors in [20] have proposed a *gossip*-based approach where each node decides to forward a message to another node based on some probability. They showed that this technique could significantly reduce the number of routing messages sent. Gossip is essentially an epidemic algorithm, where neighbors are chosen probabilistically to propagate the information in the same way as an infection spreads in a susceptible population.

In the gossip protocol a source sends the route request with probability 1. When a node first receives a route request, with probability p it broadcasts the request to its neighbors, and with probability $1 - p$ it discards the request; if the node receives the same route request again, it is discarded. Thus, a node broadcasts a given route request at most once.

The problem with gossip is that if the source has very few neighbors, then the nodes will not gossip and it would die out. Basically, from the epidemiological standpoint we say that the phase transition did not happen and the propagation collapsed. In order to circumvent this problem, the authors modify gossip so that each node forwards with probability 1 for the first k hops before continuing to gossip with probability p. The modified protocol is called the GOSSIP1(p, k) protocol.

The performance study of the gossip protocol in finite networks reveals several important results. As expected, the location of the source node does not affect the fraction of the source node receiving the messages. However, it does affect the number of executions in which the gossip dies out. The number of executions in which the gossip does not die out is higher for a more central node, and lower for a corner node. The authors observe that lowering the probability significantly changes the fraction of executions in which all nodes and no nodes get the message.

The authors suggest a few optimization techniques to the basic gossip protocol. In many cases, a gossip protocol may be run in conjunction with other protocols. If the other protocols maintain fairly accurate information regarding a nodes neighbors, GOSSIP1 can make use of this information effectively, by a simple optimization. In a random network, the number of neighbors of a node might not be very high. In such a case, the gossip protocol might not propagate the information and die out. To overcome such a situation, the authors proposed that the gossip probability at a node could be a function of its degree, where nodes with lower degree gossip with higher probability. The modified protocol has four parameters, p_1, k, p_2, and n. As in GOSSIP1, p_1 is the main gossip probability and k is the number of hops with which gossiping starts with probability 1. The new features are p_2 and n; the idea is that the neighbors of a node with fewer than n neighbors gossip with probability $p_2 > p_1$. Thus, if a node has fewer than n neighbors, it would instruct its neighbors to broadcast with probability p_2 rather than p_1. The modified protocol is called GOSSIP2 $(p_1; k; p_2; n)$. GOSSIP2 has significant impact in topologies that are random rather than regular.

However, GOSSIP1 and GOSSIP2 might suffer a premature death because the probability is low. In order to detect whether the gossip is dying out, a node might monitor the number of messages it is getting from its neighbors. If a node x has n neighbors and the message does not die out, then it would expect that all of its neighbors would get the message, and, if the gossip probability is p, it should get roughly pn messages from its neighbors. If it gets significantly fewer than pn messages within a reasonable time interval, then this is a clue that the message is dying out. The authors have proposed a modification to resolve this issue. If a node with n neighbors receives a message and does not broadcast it, but then does not receive the message from at least m neighbors within a reasonable timeout period, it broadcasts the message to all its neighbors. If m is chosen too large, then there may be too many messages. The experimental results show that the most significant performance improvement could be obtained with $m = 1$. Thus, in GOSSIP3 (p, k, m), if a node that originally did not broadcast a received message but did not get the message from at least m other nodes within some timeout period, immediately broadcasts the message after the timeout period.

5.1.1 Geographic Gossip for Efficient Aggregation

Gossip algorithms have also been used for data aggregation in sensor networks. Their forte is their simplicity in approach. However, in their basic form they may waste significant energy by essentially passing around redundant information. The authors in *Geographic Gossip* [39] propose an alternative gossiping scheme, thats exploits geographic information.

In a network of n sensors, a basic solution to the *averaging problem*, i.e., to compute the average of all n sensor measurements, is based on the *Gossip* algorithms where each node randomly picks a one-hop neighbor and exchange their current values. This is performed in an iterative fashion and ultimately all nodes converge to the global average in a distributed manner. The key issue here is the number of iterations it takes for such a gossip algorithm to converge to a sufficiently accurate estimate. Recent works [40], [41], [42], [43], [44] have dealt with variants of this problem. The convergence time of this algorithm is closely linked with the mixing time of the Markov Chain defined by a weighted random graph on the network. In [41], the authors showed how to optimize the neighbor selection probabilities for each node in order to find the fastest mixing Markov chain. However, for sensor network graphs, even an optimized gossip algorithm can result in excess energy consumption. The authors of Geographic Gossip exploit geographic information to build a completely randomized and distributed algorithm that requires substantially less communication. The idea is to include geographic routing to gossip with random nodes far away in the network. Empowered with geographic knowledge, this protocol succeeds in quickly diffusing information everywhere in the network and thus computes the average faster than the standard nearest neighbor gossip.

5.1.2 Smart Gossip

The authors of *Smart Gossip* [45] propose an adaptive form of gossiping in sensor networks. They propose techniques by which a gossip based protocol can automatically and dynamically adapt to the network topology. Smart Gossip copes well with wireless losses and unpredictable node failures that affect network connectivity. The adaptivity of the gossiping strategy also extends itself to provide reliability for disseminating messages. The authors argue that existing gossip strategies are mostly static, since there is a fixed probability for transmitting the received information. There are a few variants of the gossip protocol which are adaptive. Haas et al [20] proposed an adaptive form of gossip which chooses its probability, based on the number of neighbors. The authors of Smart Gossip argue that simply choosing gossip probabilities based on the number of neighbors is not correct. For example, in Fig. 6 which is a subgraph of a random topology, node C has a high degree and therefore its

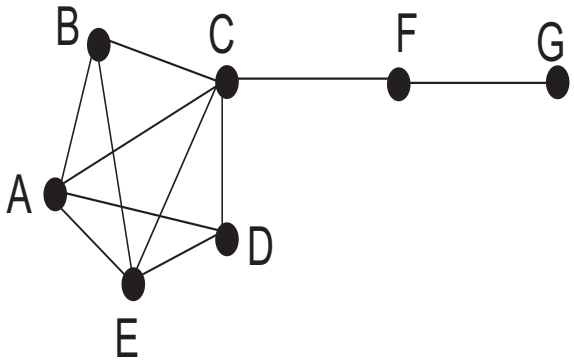

Figure 6. Example illustrating the argument for SmartGossip

probability of gossiping would be low. This could adversely affect the reception of the information at node F

which is solely dependent on C for receiving messages originating at any node to the left of C.

From the point of view of gossip percolation, the authors extract out the notion of dependence between a node X and a subset of its neighbors. These dependencies give birth to parent child relationships between neighbors based on the direction in which the gossip can travel probabilistically. This dependency graph is just logical and also probabilistic in nature. In other words, node X does not depend on any particular parent, Y, to receive the gossip. Instead, it depends on a group of nodes, expecting at least one member of this group to probabilistically deliver the gossip to it. The gossip probabilities chosen at each node is therefore a function of the group size.

Based on this intuition, nodes promiscuously overhear broadcast messages and extract information by applying simple rules and thereby deduce whether the sender of the message is a parent, child or a sibling. A child node, on identifying its parent set, calculates the probability with which it thinks its parents are required to send, and announces this probability by piggybacking it on every gossip it forwards. A parent node overhears such announcements and assumes its gossip probability to be the maximum of all the announced probabilities.

5.2 Epidemic Routing

The authors in [4] introduce epidemic routing for partially connected ad-hoc networks where pair-wise exchanges of messages among mobile hosts in a random manner ensure the eventual message delivery to the destination node. The prominent goals of Epidemic Routing are to: i) maximize message delivery rate, ii) minimize message latency, and iii) minimize the total resources consumed in message delivery.

Existing ad-hoc routing protocols assume that there is a connected path from source to destination. However, with the emergence of short-range wireless communication environments e.g., Bluetooth [16] and the wide area over which such networks are deployed, this assumption is not always a realistic one. Unfortunately, the current ad hoc routing protocols are heavily dependent on consistent network connectivity to deliver packets between the source and the destination and generally fail in the presence of network partitions. At the same time, several applications based on a mobile sensor network exist, where there are frequent and numerous formations of network partitions.

In [4], the authors develop techniques for delivering application data with a high probability even when there is never a fully connected path between the source and destination. The main essence of their approach is to distribute data to connected hosts of the network, whom they call *carriers*, and depending on node mobility, the carriers can establish contact with other connected portions of the network. Through such transitive transmission of data, messages have a high probability of eventually reaching their destination. However, with basic random forwarding, the data might be transmitted to a large number of carrier hosts other than the destination which is not desirable. Since the overall goal of epidemic routing is not just to maximize message delivery rate and minimize message delivery latency, but also to minimize the aggregate system resources consumed in message delivery, the authors circumvent this problem to a reasonable extent by placing an upper bound on the message hop count and per-node buffer space (the amount of memory devoted to carrying other hosts messages). Their results show that Epidemic Routing is able to successfully deliver messages to the destination nodes where existing ad hoc routing protocols fail because of limited node connectivity.

Although the authors of this work do not explicitly use the mathematical formulations of an epidemic model they essentially follow the same principles of the model. The contact rate between carriers and destination or intermediate connected nodes is dependent on the mobility pattern of the carriers. However, since the notion is to route and not broadcast the message, the authors successfully constrain resources at nodes to restrict the number of messages a host is willing to carry on behalf of other hosts.

5.3 Epidemiology and Mobile Ad-hoc Networks

Information diffusion in Mobile Ad-hoc Networks (MANET) has been an area where epidemiological modeling concepts fit naturally. As mentioned earlier, several modeling formulations in epidemiology assume a homoge-

neously mixing population where each infected individual has an equal probability of having contact with any susceptible individual. Scenarios which fit this assumption can borrow the differential equation based formulations popular in epidemiology. Information diffusion in MANETs fit very closely in this model. Given random mobility model, its a fair assumption that the nodes can homogeneously mix. As a result this phenomenon could be aptly modeled based on the differential rate equation formulations. This has been done by the authors in [26]. Based on a simplistic S-I-S model, the authors have simplistically modeled the spread of information in a MANET. They showed that the information dissemination can be more or less accurately described by the infection rate of the model. They derived expressions which show the change of infection rate based on the node densities.

In [35], the authors address the issue of how to disseminate relevant information to mobile agents within a geosensor network. In their work, the authors propose an environment for simulating information dissemination strategies in mobile ad hoc geosensor networks. A geosensor network is defined as a sensor network that monitors phenomena in geographic space [36]. In the context of geosensor networks, the authors provide a decentralized location based service that is able to disseminate relevant geospatial information to spatially dispersed mobile users that form a mobile ad-hoc geosensor network. The authors explore the precise nature of efficient information dissemination strategies based on localized communication between agents in a geosensor network. Specifically, they are concerned with mobile location-aware agents who are able to sense information about their immediate geospatial environment and communicate with other agents in their neighborhood. The authors distinguish between three different strategies. The first strategy, *Flooding*, is where each geosensor node that encounters an event or receives a message about an event passes on the information to every other node within its communication range. The second approach is referred to as an *Epidemic*, in which each node only informs n other agents about the events. In the third approach which is *location-constrained*, information is only passed on in proximity to the event, and then discarded.

In [37], the authors propose a document oriented model for information dissemination in mobile ad-hoc networks. The problem of routing messages in disconnected or partially connected mobile ad-hoc networks has been dealt by previous works like [4], [38]. The main contribution in [37] is the implementation of a service for document dissemination in ad-hoc networks and then using this service as a building block for application level services.

Any document that is sent in the network is cached as long as possible by as many devices as possible, so that it can remain available for those devices that could not receive it at the time it was sent originally. Other than providing a caching system where documents can be maintained in mobile devices, their service also provides facilities for document advertisement, document discovery, and document transport between neighboring devices. A device can periodically advertise to its neighbors about the documents stored in its cache. It can also search for specific documents in its neighborhood, and either push documents toward or pull documents from its neighbors.

In [47], the authors propose a middleware for a controlled epidemic style dissemination for mobile ad-hoc networks. Since traditional middleware primitives offer very little information dissemination mechanisms and epidemic algorithms have hardly been used to control the spreading of information depending on the desired reliability and network structure, the authors present a mobile ad hoc network middleware which uses epidemic-style information dissemination techniques to tune the reliability of the communication.

The authors argue that existing epidemic algorithms have little control on the information dissemination process and much of it is based on experimental results and not on any analytical model. In other words, the information spread cannot be accurately tuned in order to reach only a desired percentage of the hosts.

The authors, therefore, propose algorithms that rely on epidemic models and take into account the underlying network structure. They design middleware interfaces that allow programmers to set the reliability for unicasting and anycasting with a high degree of accuracy. The middleware would have primitives for epidemic dissemination and would take as control inputs the percentage of hosts to disseminate the information to. The authors use the infectivity, which is the probability of being infected by a neighboring host, to control the reliability of the probabilistic unicast. Thus, given an expected reliability value, the middleware is able to calculate the infectivity

accurately in order to obtain an infection rate proportional to the total number of hosts in the network. For constructing the analytical model the authors adopt the simple S-I-S model of epidemiological spread to model the information dissemination in a MANET. For the analytical model, the authors assume homogeneous mixing of the nodes and the infectivity of a single host, per message is constant. Using the average node degree and the probability of infection, the authors calculate the infection rate. Based on its calculation the authors depict the epidemic spread algorithm which is executed periodically.

6 Epidemic Models of Malicious Code Propagation

Computer worms have recently emerged as one of the most critical threats against information confidentiality, integrity and service availability. Host machines in the Internet have repeatedly revealed their susceptibility to malicious intrusions like worms that have compromised millions of vulnerable hosts at an extremely fast pace [13], [14]. Given the threat of virus and worm propagation in wireless networks quite real, a few recent works have tried to focus on this idea and successfully utilized the concept of epidemic theory to model the spread of worms in wireless ad-hoc and sensor networks.

In [28], the authors discuss about the epidemic model of virus spreading in mobile environments. Given the increasing rise in the usage of mobile devices, it is a matter of time before viruses propagating over the air interface would be a major menace. Already, there are several viruses and worms that spread over the air. For example, the Brador virus [29] infects Pocket PCs running Windows CE, and by installing a backdoor it allows a remote attacker access to the device. The Cabir worm [30] infects cell phones running the Symbian operating system. Identifying these examples, the authors of [28] stress on several important factors like movement of devices and the geographic locations while formulating epidemic models for virus spread in mobile environments. In their model which they call *probabilistic queuing*, the authors investigate the behavior of malicious code that spread via proximity-based point-to-point wireless links. They point out the drawbacks that existing epidemiological modeling of similar processes in mobile environments have, like ignoring the node velocity and the non-homogeneous connectivity distributions that often arise in mobile environments. The Kephart-White (KW) Model [31] assumes a homogeneously connected topology and the network is represented by a single parameter, i.e., the average node degree. However, mobile environments are too dynamic in order for this model to fit. The fact that the KW model only considers mean connectivity, discards useful information when the underlying distribution has significant variance which is normal in a mobile environment. Furthermore, the velocity is an important factor that influences the way a virus can spread among mobile nodes. The authors incorporate this parameter in their model, and come up with a new epidemic threshold value when the virus spread reaches endemic state. In the KW model which ignores node mobility, this threshold is crossed when the infecting rate is greater than the curing rate. In their model, the authors incorporate the mobility model in their derivation of a node's degree distribution which leads to a new value for the epidemic threshold.

Several other works discuss various aspects of vulnerabilities of sensor and mobile ad-hoc networks in the light of epidemic theory.

6.1 TWPM

In [15], the authors develop a *topologically-aware worm propagation model* (TWPM) for wireless sensor networks. An important strategy effectively employed by many recent worms (e.g., CodeRed v2) is *localized scanning*. The local scanning worms after compromising a host machine, instead of scanning a fixed IP address space, scan neighboring hosts with a higher probability. This strategy has proven to be quite effective since the presence of a single vulnerable host implies that other hosts on the same network would also be vulnerable with a high probability.

Since general routing strategies in sensor networks have each sensor maintain a neighbor list, this procedure of

localized scanning could be very effective for a virus spreading in a sensor network. Moreover, since the worm under consideration employs (next-hop) information from a sensor to infect other sensors, the authors refer to it as a topologically-aware worm. Based on the S-I-S model of epidemic spread, the authors have constructed a differential equation based worm propagation model in sensor networks. Apart from simultaneously capturing both time and space propagation dynamics, TWPM incorporates physical, MAC and network layer considerations of practical sensor networks.

Dividing the sensor network into equal sized segments, and using a constant rate of infection, the authors have arrived at a closed form expression for the number of infectives at time t that also successfully captures the spatial information in terms of the segment coordinates.

6.2 Compromise Propagation in secure sensor networks

TWPM modeled the worm propagation process using a differential equation based approach. However, generally in a static network, e.g., a sensor network, the differential equation based approach is not feasible since it assumes a homogeneous mixing of the susceptible nodes and the infected nodes. In such a scenario, a network or graph theoretic modeling technique is much more suitable to capture the propagation process. One such novel work has been done by the authors in [3] where they model the process of how a compromised node in a sensor network gradually compromises other nodes and eventually the whole network.

The authors assume the nodes in the sensor network to be uniformly randomly deployed in an area and securely communicating among themselves. By secure communication, the authors assume a secret shared key based communication paradigm. They assume a prior random key distribution technique to have distributed secret keys to each node using which they communicate with each other. Given such a securely communicating sensor network, the authors study how an adversary who has captured one or two sensor nodes and extracted their secret keys, can possibly propagate the compromise of nodes to the whole network.

When a node is captured and its keys are known by the attacker, secure communication can be established with neighboring nodes with which the captured node shares keys. Being able to securely communicate with its neighbor, the node with the malicious code can easily attain its susceptible neighbor's trust and pass on the malicious code to the latter. Once it has passed to the susceptible neighbor, the authors assume that the malicious code has the ability to acquire the secret keys of the new node. This is when the new node also becomes infected and results in the propagation of malicious code. This process continues until the whole network gets compromised.

By constructing a random graph model of the key sharing overlay graph of the sensor network, the authors present the compromise propagation model as a poisson process with a mean which is dependent on the *infection probability* and the *infectivity duration* at each node. The propagation process was expressed by a *transmissibility* factor of the infection and it was basically analogous to the bond occupation probability on the graph representing the key sharing network. The size of the epidemic was equivalent to the size of the giant component formed with edge existence probability defined by the transmissibility of the compromise process.

The main focus of their work was to identify the phase transition points of the process when it attains epidemic proportions. They studied the effects of the compromise propagation under two scenarios, viz., without node recovery and with node recovery. In the event of a compromise, the network may attempt to recover the particular node. Recovery might be realized in several possible ways. For example, the keys of the nodes might be revoked and the node may be given a fresh set of secret keys. In this context, key revocation, which refers to the task of securely removing keys that are known to be compromised, has been investigated as part of the key management schemes, for example in [46]. Moreover, recovery can also be achieved by simply removing the compromised node from the network, for example by announcing a blacklist, or simply reload the nodes programs. More sophisticated methods may include immunizing a node with an appropriate antivirus patch that might render the node immune from the same virus attack. Regardless, in their analysis, the authors studied virus spreading under the two cases respectively depending on whether a node can be recovered or not.

Since, contrary to the differential rate equation based modeling methods, the graph theoretic model does not capture the temporal effects of an epidemic, the authors captured the temporal dynamics of the propagation process using simulation techniques.

7 Conclusion and Open Issues

In this chapter we have delved into various epidemiological models and protocols employed in wireless ad hoc and sensor networks. Starting from data dissemination and gossip protocols to security issues in sensor networks such as propagation of compromise of sensor nodes, we observe that there have been several works inspired by this powerful concept of epidemic theory. The density and scale of a sensor network coupled with the objective of a one-to-many data transfer from a few nodes to the rest of the network, unleash the efficiency with which this theory can effectively model and provide solutions to several problems in ad hoc and sensor networks. In this chapter, we have tried to touch most of the salient contributions that have adopted this theory and provide a concise compilation under various categories of sub-classifications.

Among the open issues related to the modeling of sensor networks, we find that not many protocol models in sensor networks have been proposed with the nodes being mobile. With mobile nodes, the dynamics of the network and its properties like connectivity keep changing continuously, thus making it more difficult to capture in an analytical closed-form model. For example, the analysis and performance study of broadcast protocols in a mobile sensor network environment still requires considerable research. Another important aspect that needs to be dealt with while modeling a sensor network protocol is the node deployment scenario. Apart from the straightforward uniform random deployment of the sensor nodes, there could be other distributions for node deployment. A popular one among them is deploying nodes in packets such that the resident points of nodes from each packet forms a two dimensional gaussian distribution about the dropping point. Given such a position distribution of the sensor network, epidemic modeling of dissemination or broadcast protocols would have to deal with the change in degree distribution dependent on the location of a node in the network.

8 Exercise

- How is epidemic modeling in sensor networks different from that used for the Internet?
- What is the Basic Reproductive Number in Epidemiology? How is it different for Scale Free Networks from other networks?
- How does a homogeneously mixing population and a heterogeneously mixing one affect the mathematical formulation of the spreading process in Epidemiology?
- How does mobility of the network nodes change the mathematical formulation of an epidemic model of the network?
- What are the important parameters that determine the choice between a continuous and a discrete time epidemic model of a network?
- What are the important issues while epidemic modeling of malware spread in a mobile environment? How do the location of a mobile device and the time of the day affect the model?

References

[1] R. M. Anderson and R. M. May, Infectious Diseases of Human: Dynamics and Control (Oxford Univ. Press, Oxford, 1991).

[2] A. Woo, S. Madden, and R. Govindan. “Networking Support for Query Processing in Sensor Networks”. *Comm. of the ACM*, 47(6):4752, 2004.

[3] Pradip De, Yonghe Liu, and Sajal K. Das, ”Modeling Node Compromise Spread in Sensor Networks using Epidemic Theory”, In *World of Wireless, Mobile and Multimedia Networks*, WoWMoM 2006.

[4] A. Vahdat and D. Becker. Epidemic routing for partially-connected ad hoc networks. Duke Technical Report CS-2000-06, July 2000.

[5] R. Pastor-Satorras and A. Vespignani, “Epidemic dynamics and endemic states in complex networks”, Phys. Rev. E, 63 (2001), art. no. 066117.

[6] R. Pastor-Satorras and A. Vespignani, “Epidemic spreading in scale-free networks”, Phys. Rev. Lett., 86 (2001), pp. 32003203.

[7] R. M. May and A. L. Lloyd, “Infection dynamics on scale-free networks”, Phys. Rev. E, 64 (2001), art. no. 066112.

[8] I. de S. Pool and M. Kochen, “Contacts and influence”, Social Networks, 1 (1978), pp. 148.

[9] M. E. J. Newman, “Spread of epidemic disease on networks”, Phys. Rev. E, 66 (2002), art. no. 016128.

[10] C. Moore and M. E. J. Newman, “Epidemics and percolation in small- world networks”. Phys. Rev. E 61, 5678-5682 (2000)

[11] P. Grassberger, “On the critical behavior of the general epidemic process and dynamic percolation”, Math. Biosc. 63 (1983) 157.

[12] C. Griffin, R. Brooks, “A note on the spread of worms in scale-free networks”, In IEEE Transactions on Systems, Man and Cybernetics, Vol 36, Issue 1, Feb, 2006.

[13] S. Staniford, V. Paxson, and N. Weaver, “How to Own the Internet in Your Spare Time,” Usenix Security Symposium, 2002.

[14] C. Shannon and D. Moore, “The Spread of the Witty Worm,” IEEE Security & Privacy, vol. 2, no. 4, July/August 2004.

[15] S. A. Khayam and H. Radha, “A Topologically-Aware Worm Propagation Model for Wireless Sensor Networks”, In IEEE ICDCS International Workshop on Security in Distributed Computing Systems SDCS, 2005.

[16] J. Haartsen, M. Naghshineh, J. Inouye, O. J. Joeresson, and W. Allen. “Bluetooth: Vision, Goals, and Architecture”. ACM Mobile Computing and Communications Review, 2(4):3845, October 1998.

[17] M. Akdere, C. C. Bilgin, O. Gerdaneri, I. Korpeoglu, O. Ulusoy, U. Cetintemel, “A comparison of epidemic algorithms in wireless sensor networks”, Computer Communication, 2006.

[18] C. E. Perkins and E. M. Royer, “Ad-hoc on-demand distance vector routing”, In 2nd IEEE Workshop on Mobile Computing Systems and Applications, pages 90100, February 1999.

[19] D. B. Johnson and D. A. Maltz. “Dynamic Source Routing in Ad Hoc Wireless Networks”, Kluwer Academic Publishers, 1996.

[20] Z. J. Haas, J. Y. Halpern, and L. Li, “Gossip-Based Ad Hoc Routing”, In INFOCOM 2002.

[21] P. Levis, N. Patel, D. Culler, S. Shenker, "Trickle: a self-regulating algorithm for code maintenance and propagation in wireless sensor networks", in the First USENIX/ACM Symposium on Network Systems Design and Implementation (NSDI), 2004.

[22] P. Levis, and D. Culler, "The firecracker protocol". In Proceedings of the 11th ACM SIGOPS European Workshop, Leuven, Belgium, 2004.

[23] D. Ganesan, B. Krishnamachari, A. Woo, D. Culler, D. Estrin, and S. Wicker, "An Empirical Study of Epidemic Algorithms in Large Scale Multihop Wireless Networks" Intel Research, Berkeley Technical Report IRB-TR-02-003, March, 2002.

[24] J. Kulik, W. Rabiner, H. Balakrishnan, Adaptive protocols for information dissemination in wireless sensor networks. Proceedings of the Fifth ACM/IEEE Mobicom Conference, Seattle, WA, August 1999.

[25] J. W. Hui and D. Culler, "The dynamic behavior of a data dissemination protocol for network programming at scale", 2nd International Conference on Embedded Networked Sensor Systems, 2004

[26] A. Khelil, C. Becker, J. Tian, and K. Rothermel, "An Epidemic Model for Information Diffusion in MANETs", in MSWiM, 2002.

[27] A. Beaufour, M. Leopold, and P. Bonnet, "Smart-Tag Based Data Dissemination", in WSNA, 2002.

[28] J. W. Mickens and B. D. Noble, "Modeling Epidemic Spreading in Mobile Environments", in WiSE, 2005.

[29] R. Wong and I. Yap. Security Information: Virus Encyclopedia: WINCE BRADOR.A: Technical Details, 2004. Trend Micro Incorporated.

[30] P. Ferrie, P. Szor, R. Stanev, and R. Mouritzen. Security Response: SymbOS.Cabir, 2004. Symantec Corporation.

[31] J. Kephart and S. White. Directed-graph epidemiological models of computer viruses. In Proceedings of the IEEE Computer Symposium on Research in Security and Privacy, pages 343359, May 1991.

[32] S. S. Kulkarni and M. Arumugam, "INFUSE: A TDMA based Data Dissemination Protocol for Sensor Networks", Technical Report MSU-CSE-04-46, Department of Computer Science, Michigan State University, November 2004.

[33] S. S. Kulkarni and L. Wang. "MNP: Multihop network reprogramming service for sensor networks", In Proceedings of the 25th International Conference on Distributed Computing Systems ICDCS, pages 716, June 2005.

[34] V. Naik, A. Arora, P. Sinha, and H. Zhang. "Sprinkler: A reliable and energy efficient data dissemination service for wireless embedded devices", In Proceedings of the 26th IEEE Real-Time Systems Symposium, December 2005.

[35] S. Nittel, M. Duckham, and L. Kulik, "Information Dissemination in Mobile Ad Hoc Geosensor Networks", Third International Conference on Geographic Information Science, 2004.

[36] S. Nittel, et al, "Report from the first workshop on geo sensor networks", ACM SIGMOD Record 33 2004.

[37] F. Guidec, H. Roussain, "Asynchronous Document Dissemination in Dynamic Ad Hoc Networks" In Second International Symposium on Parallel and Distributed Processing and Applications (ISPA), 2004.

[38] A. Lindgren, A. Doria, and O. Schelen. “Probabilistic Routing in Intermittently Connected Networks”. In Proceedings of the Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2003), June 2003.

[39] A. G. Dimakis, A. D. Sarwate, and M. J. WainWright, “Geographic Gossip: Efficient Aggregation for Sensor Networks”, In Fifth International Symposium on Information Processing in Sensor Networks (IPSN) 2006.

[40] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. “Analysis and optimization of randomized gossip algorithms”, In Proceedings of the 43rd Conference on Decision and Control (CDC 2004), 2004.

[41] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. “Gossip algorithms : Design, analysis and applications”, In Proceedings of the 24th Conference of the IEEE Communications Society (INFOCOM 2005), 2005.

[42] J.-Y. Chen and D. X. G. Pandurangan. “Robust aggregates computation in wireless sensor networks: Distributed randomized algorithms and analysis”. In Fourth International Symposium on Information Processing in Sensor Networks (IPSN), 2005.

[43] R. Karp, C. Schindelhauer, S. Shenker, and B. Vocking, “Randomized rumor spreading”, In Proc. IEEE Conference of Foundations of Computer Science, (FOCS), 2000.

[44] D. Kempe, A. Dobra, and J. Gehrke, “Gossip-based computation of aggregate information”, In Proc. IEEE Conference of Foundations of Computer Science, (FOCS), 2003.

[45] P. Kyasanur, R. RoyChoudhury, and I. Gupta, “Smart Gossip: Infusing Adaptivity into Gossiping Protocols for Sensor Networks”, to appear in ICDCS 2006.

[46] H. Chan, V. D. Gligor, A. Perrig, G. Muralidharan, “On the Distribution and Revocation of Cryptographic Keys in Sensor Networks”, IEEE Transactions on Dependable and Secure Computing 2005.

[47] M. Musolesi and C. Mascolo, “Controlled Epidemic-style Dissemination Middleware for Mobile Ad Hoc Networks”, In Mobiquitous 2006.