IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 3, Ver. V (May-Jun. 2014), PP 66-70
www.iosrjournals.org

Examining basic probabilistic, time-to-live, kill time and Passive Cure problems in flooding

[1]S.I.S.Jaffarvalli, [2]Gunasekhar
[1, 2] Mtech student CSE Aits, Rajampet

Abstract: *Mobile adhoc networks are dyanamic in nature hence it is difficult to predict the toplogy of manets. basically in manets flooding is simple routjng algorithm in which every incoming packet is sent through every outgoing link except except the one it arrived on. The broadcast storm problem is so common in flooding algorithms that it has engendered a whole area of research flooding schemes over large-scale, sparse mobile networks. Traditional flooding suffers from the problem of redundant message reception, once per neighbour. Even in a reasonably connected network, the same message is received multiple times by every node, which is inefficient, wastes valuable resources, and can create contention in the transmission medium. In this paper, we focus on the evaluation of different controlled messages.*

Keywords: *MANET, flooding, Delay Tolerant Networking*

I. Introduction:

Flooding is the mechanism by which a node, receiving flooded message m for the first time, rebroadcasts once. When distinguish between flooding and broadcast, which is a transmission that is received by all nodes within transmission range of the broadcasting node. Flooding usually covers all the nodes in a network, but can also be limited to a set of nodes that is defined by a geographical area (also called geocast flooding [15]) or by the time-to-live (TTL) parameter of m. Thus, a node receiving the flooded message only rebroadcasts it if it is within the specified area or if the message's TTL is greater than 0. Research in the areas of Mobile Ad Hoc Networks (MANETs)[1],[2], disconnected sparse networks [7] and Delay Tolerant Networks (DTNs), [13],[14], has addressed some of these challenges and problems. However, despite the great contribution of these areas, many issues require further research. This can be demonstrated by the following scenario: A person is driving on a highway in Bangalore, carrying his own lap-top or PDA, and needs to send an urgent email or submit a transaction. There is no nearby connectivity (in a desert for instance). The user passes other cars, buses or trains that have other people carrying similar devices. These users can serve as relays for the email or transaction and pass it on to others. Eventually, the message reaches someone with Internet connectivity and a direct path to the destination. This scenario is an example of what we call a large-scale sparse mobile network. Most research fails to solve many problems posed by such scenarios. MANETs, for instance, focus more on networks where an end-to-end path is assumed to exist. On the other hand, partially-connected and sparse networking research usually assumes some sort of control over nodes in the network, a large degree of homogeneity, or some degree of knowledge that nodes must carry regarding other nodes in the network (e.g. the path or route a node will take). Clearly these assumptions do not hold in our scenario. DTN research comes closest to addressing the problems that arise from the scenario mentioned above. Our work complements that of the DTN community.

II. System Architecture:

There are two important assumptions made in our system**.** The first is Node Blindness, where the nodes in the network do not know any information regarding the state, location or mobility patterns of other nodes. The second is Node Autonomy, where each node has independent control over itself and its movement. The reason behind these assumptions is to closely model the real world scenarios described earlier. When driving through a sparse environment, any given node has no knowledge of other nodes that come within its range(Blindness), and it is autonomous in its movement (Autonomy). In Figure 1 we show that the nodes in the network are divided into three types. A Sender Node "S" is the node that initiates the transmission of a message to a destination in the network. A Forwarder Node "F" is any node that carries the message from the sender, or another forwarder, with the aim of relaying it to the ultimate node. Finally, the Ultimate Node "U" is basically the final destination.

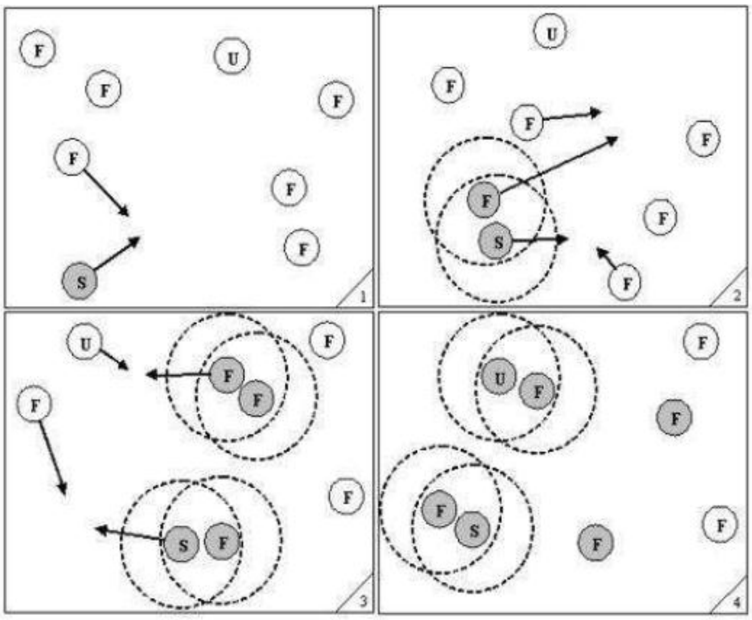


Fig. 1. Message propagation over a sparse mobile network. Shaded nodes are those Carrying a copy of the message. Arrows indicate the direction of movement.

The basic mechanism of node interaction is shown in Figure 1. The interaction of nodes is similar to that in Epidemic Routing [8], where each node continuously tries to relay the message to other nodes within range that do not already have the message. We look at an example where a sender node, S, needs to send a message. At this point, node S initiates a periodic beacon for neighbour discovery purposes, where it announces that it has a message that needs to be forwarded to a certain destination. When S comes within range of one or more forwarder nodes F (or even the ultimate node U), the beacon is received and an ack is sent to S from each node that received the beacon and does not have a copy of the message. When S receives an ack for its beacons, it simply broadcasts the message to its neighbours. Once the message is received, the forwarder node starts to act as a sender node. It sends its own beacons, and both nodes travel through the network looking for either another forwarder to pass the message on, or for the ultimate node. The message gradually propagates through the network until it eventually reaches the ultimate node. This process results in the overuse of network resources through continuous and repetitive flooding of messages. On the other hand, the advantage of this approach is the high delivery rate and small delay.

III. Speccific Controlled Flooding Schemes:

1) **Basic Probabilistic (BP):** When talking about the willingness of a node in the previous section, we implied that forwarder nodes have the same willingness as the sender node. To more closely emulate reality, however, we choose a uniform distribution probabilistic function that determines the willingness of the nodes to transmit a given message. Based on the result of this function, a forwarder may choose not to forward the message at all, forward it at half the willingness of the sender or forward it at the same level of willingness as the sender.

2**) Time-to-Live (TTL):** In this scheme, we add a time-to-live value. The TTL here determines how many times the message is forwarded before it is discarded. We add the TTL on top of the BP scheme since the BP scheme is a more realistic representation of how nodes act regarding the choice of forwarding messages.

3**) Kill Time:** Here, we add a time stamp to the message on top of the BP scheme. The time stamp is the time interval after which the message should no longer be forwarded. This is an absolute universal life-time for the message. This could be very useful if the sender node knows how long it will be disconnected. This is also a good way to set the maximum time a node should keep a message in its buffer if the times-to-send (TTS) variable of that message does not reach zero.

4**) Passive Cure**: The optimization we introduce is a Passive Cure. The idea is that once the destination (Ultimate node) receives the message, it generates a Passive Cure to \heal" the nodes in the network after they have been \infected" by the message. The ultimate node cures" the forwarder that passed the message to it by sending a cure ack instead of a normal ack that is sent when a beacon is received. Now whenever that forwarder or the ultimate destination detect any other node sending that same message, they send a cure ack to that node to prevent future retransmissions.

IV. Simulation Environment

The main goal for our simulations in this paper is to compare the performance of the schemes described in Section 3 and study their impact on our metrics. To start, we _rst describe our simulation environment. We conducted our simulations using the GloMoSim network simulator. Since we do not have real data describing our targeted scenarios, we use the random way-point model since, intuitively, it most closely approximates the scenarios we are concerned with. The node speed ranges between 20 to 35 meters per second (to get a range of almost 45 to 80 miles per hour); the rest period is between 0 and 10 seconds.

Parameter	Value Range	Nominal Value
Number of Nodes	10 to 250	100
Terrain	$10km^2$ to $50km^2$	$10km^2$
Simulated Time	1hour to 24 hours	6 hours
Transmission Range	250m	250m

Table 1. Simulation Parameters

Times-to-Send (TTS): Number of times to successfully forward a message. It ranges from 1 to 50 with a nominal value of 10.{Retransmission Wait Time (RWT): Time period after which a node tries to resend a message. It ranges from 0 sec to 500 sec with a nominal value of 50 sec.

Time-to-Live (TTL): The number of hops after which a message is discarded. It ranges from 2 to 10 with a nominal value of 7

Kill Time: The absolute time after which the message is discarded. It ranges from 1000 sec to 21600 sec (6 hours, the nominal simulation duration time)
with a nominal value of 5000 sec.

V. Experimental Results:

Before applying our probabilistic scheme, we analyze how the network acts as suming full willingness of all the nodes in the network along with enforcing some basic level of message control. This scheme is very similar to the Epidemic Routing scheme but with some control over message forwarding. We use the times-to-send (TTS) variable to represent node willingness. We investigate howvarying the TTS and the network density aect our metrics. The retransmission wait time (RWT) in these experiments is 1 sec.

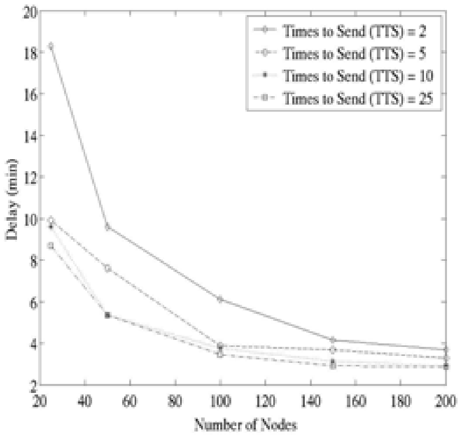


Fig. 2. Impact of changing the number of nodes and times-to-send (TTS) on the total Number of messages in the basic scheme.

After applying our basic probabilistic scheme, we examine how the network density, retransmission wait time (RWT) and times-to-send (TTS) impact ourmetrics. We assume that 25% of the nodes in the network have zero willingness, 25% have full willingness and 50% of the nodes forward the message with only half the willingness (i.e. half the TTS of the source node). Figure 3 shows the result of varying the network density while keeping the TTS set to 10. One interesting observation is the sign cant drop in terms of the total number of messages when the RWT increases, with only a corresponding small increase in delay (until RWT reaches 100). With a small RWT, the message spreads rapidly through the network, and in a very short time, many forwarders

are actively trying to send the message. As the RWT increases, the message does not initially reach as many forwarders. This results in a significant drop in the number of messages. Also, with a large RWT, the TTS is not consumed as quickly, so a forwarder node stays in forwarder mode longer, thus rejecting
receipt of the same message. With a small RWT, the TTS is quickly consumed.
Because the nodes do not keep any state or cache, they are ready to receive the

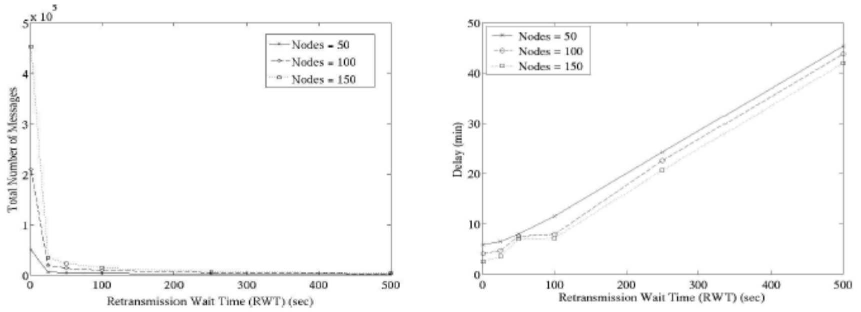
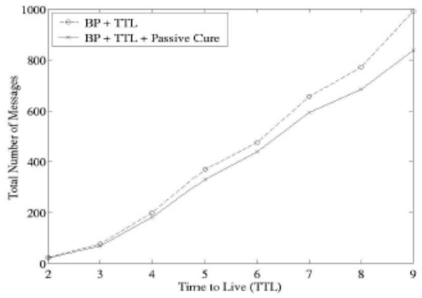


Fig. 3. Impact of changing the retransmission wait time (RWT) and number of nodes on the total number of messages (a) and overall delay (b).

Time-to-Live (TTL)

With the improvement in the basic probabilistic scheme, we build the other schemes on top of it to see their impact on our metrics. Figure 4 shows the impact of adding TTL only and adding TTL + Passive Cure, to the basic probabilistic (BP) scheme. We discuss the Passive Cure results later and here we focus on the TTL. The result we observe in Figure 4 is the large decrease in the total number of messages sent in the network (note the scale is significantly smaller than in qures 2(a) and 3(a)). This is because the TTL puts a limit on the number of times a message is forwarded. Previously, messages endlessly propagated throughout the network with no limit other than the nodes' willingness



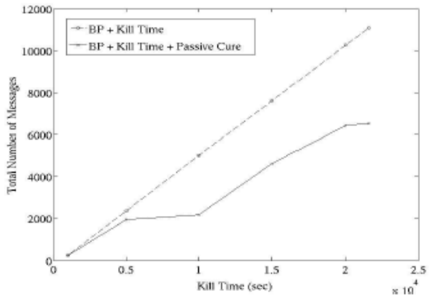


Fig. 4. Impact of adding TTL and Pas-sive Cure to BP. Fig. 5. Impact of adding Kill Time andPassive Cure to BP Kill Time

Figure 5 shows the impact of adding kill time only, and adding kill time + Passive Cure, to the basic probabilistic (BP) scheme. Here we focus on the kill time results, leaving the Passive Cure for later.The kill time scheme introduces another improvement over the basic probabilistic (BP) scheme alone. The use of a universal time after which a message is discarded certainly stops message transmission and propagation. Figure 5 shows how the total number of messages increases as the kill time is increased. On the other hand, the overall delay (not shown due to space limitation) remains constant at 10 mins, unless the kill time is set too small that some of the messages do not make it to the destination.

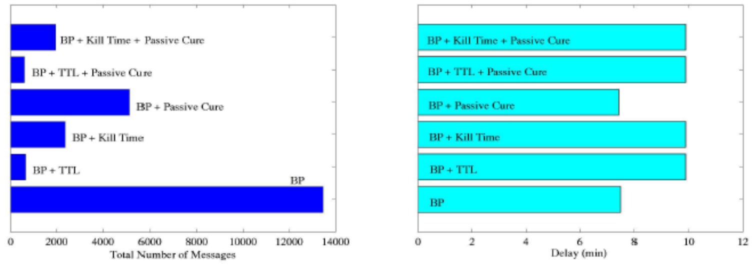


Fig. 6. Comparing the impact of the controlled flooding schemes on the total number of messages (a) and the overall delay (b).

Comparing Schemes

When looking at Figure 6(a), we observe that the Basic Probabilistic scheme by itself is the most expensive, while the basic probabilistic (BP) + TTL + Passive Cure is the least expensive in terms of the total number of messages sent in the network. Note that the basic probabilistic (BP) scheme already performs better than the basic fooding technique (which is analogous to Epidemic Routing). Figure 6(b) shows that most of the schemes have similar overall delay. The advantage is for the basic probabilistic (BP) and basic probabilistic (BP) + Passive Cure schemes. The reason for this is that the complexity introduced bythe TTL or the kill time results in an overall increase in delay. However, a two minute increase, with a corresponding large decrease in number of messages and beacons sent, is certainly acceptable. Figure 6 summarizes the general tradeo_s between the schemes we introduce.For example, adding the Passive Cure to the basic probabilistic (BP) scheme saves about 60% of the total number of messages, with no increase in the overall delay. When adding the TTL to the basic probabilistic (BP), the total number of messages is reduced by more than 90%, while the overall delay increased by only two minutes. This increase in delay does not notably affect most of the applications we envision for this work.

VI. Conclusion

The specific schemes it can be examined are basic probabilistic, time-to-live, kill time and Passive Cure. We study the impact of these schemes on network effiency and overall message delivery delay. Simulations show that for a given sparse mobile network, the schemes signicantly reduce the total number of messages and beacons sent in the network. This occurs with either no increase or only a small affordable increase in the overall message delay. Our future work includes the transmission of a message to multiple destinations. Also, since fooding based schemes do not perform well in dense environments, we need to add measures to help the nodes modify their behavior when they enter densely populated areas.

References

[1]. Johnson, D., Maltz, D.: Dynamic source routing in ad hoc wireless networks. In:Mobile Computing. Volume 353. (1996)

[2]. Perkins, C.: Ad-hoc On-Demand Distance Vector Routing. In: Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA (1999) 90

[3]. Haas, Z., Pearlman, M.: The Performance of Query Control Schemes for Zone Routing Protocol. In: ACM SIGCOMM, Vancouver, Canada (1998)

[4]. Royer, E., Toh, C.: A Review of Current Routing Protocols for Ad-hoc Mobile Wireless Networks. IEEE Personal Communications Magazine 6 (1999)

[5]. Perkins, C., Bhagwat, P.: Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In: ACM SIGCOMM, London, England (1994)

[6]. Ko, Y.B., Vaidya, N.: Location-Aided Routing (LAR) in Mobile Ad hoc Networks.

[7]. In: ACM MobiCom, Dallas, TX (1998) Li, Q., Rus, D.: Sending Messages to Mobile Users in Disconnected Ad-Hoc Wire-less Networks. In: ACM MobiCom, Boston, MA (2000) 44-55

[8]. Vahdat, A., Becker, D.: Epidemic Routing for Partially Connected Ad Hoc Net-works. Technical Report CS-200006, Duke University (2000)

[9]. Juang, P., Oki, H., Wang, Y., Martonosi, M., Peh, L., Rubenstein, D.: Energy- E_cient Computing for Wildlife Tracking: Design Tradeos and Early Experiences With ZebraNet. In ASPLOS (2002)

[10]. Shah, R., Roy, S., Jain, S., Brunette, W.: Data MULEs: Modeling a Three-Tier Architecture for Sparse Sensor Networks. In IEEE SNPA Workshop (2003)

[11]. Zhao, W., Ammar, M.: Proactive Routing in Highly-Partitioned Wireless Ad Hoc Networks. In: The 9th IEEE International Workshop on Future Trends of Distributed Computing Systems, San Juan, Puerto Rico (2003)

[12]. Zhao, W., Ammar, M., Zegura, E.: A Message Ferrying Approach for Data Delivery in Sparse Mobile Ad Hoc Networks. In: MobiHoc, Tokyo, Japan (2004)

[13]. Fall, K.: A Delay-Tolerant Network Architecture for Challenged Internets. In:ACM SIGCOMM, Karlsruhe, Germany (2003)

[14]. Jain, S., Fall, K., Patra, R.: Routing in a Delay Tolerant Network. In: ACM SIGCOMM, Portland, Oregon (2004)

[15]. Cerf, V., Burleigh, S., Hooke, A., Togerson, L., Durst, R., Scott, K., Travis, E.,Weiss, H.: Interplanetary Internet (IPN): Architectural De_nition. IETF Internet Draft, draft-irtf-ipnrg-arch-00.txt (work in progress) (2001)