Epidemic Routing Protocol Implementation in ns-3

Mohammed J.F. Alenazi*§, Yufei Cheng*, Dongsheng Zhang*, and James P.G. Sterbenz*†‡
*Information & Telecommunication Technology Center, Department of Electrical Engineering and Computer Science The University of Kansas, Lawrence, KS 66045, USA
§College of Computer & Information Sciences
King Saud University, Riyadh, Saudi Arabia
†School of Computing and Communications and InfoLab21
Lancaster University, Lancaster LA1 4WA, UK
‡Department of Computing
The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong
{malenazi, yfcheng, dzhang, jpgs}@ittc.ku.edu
www.ittc.ku.edu/resilinets

ABSTRACT

Routing protocols play a significant role in the overall performance of ad-hoc wireless networks. Several routing protocols have been proposed for ad hoc environments. Any new proposed protocol should be compared with other routing protocols to show its performance under several scenarios. Epidemic routing was one of the first routing schemes proposed for DTNs (delay-tolerant networks). In this paper, we present our implementation of the epidemic routing protocol in the ns-3 simulator. We analyse its performance and compare with the previous ns-2 implementation. Our analysis conforms the results of the previous ns-2 implementation. Moreover, we compare our epidemic implementation to other MANET routing protocols in a delay tolerant environment and we show that epidemic routing outperforms other MANET routing protocols in terms of packet delivery at the expense of overhead and delay.

Categories and Subject Descriptors

I.6 [**Simulation and Modeling**]: General, Model Development, Model Validation and Analysis; C.2.2 [**Computer-Communication Networks**]: Network Protocols—*routing protocols*

General Terms

Implementation, Analysis, Testing, Verification

Keywords

Epidemic routing protocol, MANET, mobile ad hoc networking, delay tolerant networking, DSR, AODV, DSDV, OLSR, ns-3 simulation

WNS3 2015, May 13 2015, Barcelona, Spain
 ISBN 978-1-4503-3375-7/15/05 ... $15.00
DOI: http://dx.doi.org/10.1145/2756509.2756523.

1. INTRODUCTION

With the exponential growth of wireless devices, commercial and educational wireless applications have become more common in our daily lives [1]. Wireless mobile ad hoc networks can be established in a distributed and self-organised manner without preexisting infrastructure. Conventional ad hoc routing techniques assume that there is always an end-to-end path available for node pairs to communicate with each other. In reality, dynamically changing topologies could cause the network to frequently partition. In this situation, many packets might not be delivered due to the lack of a complete path between source and destination while using traditional MANET (mobile ad hoc network) routing protocols. This requires a protocol that can constantly respond to newly available partial paths. Furthermore, the available paths between source and destination might be extremely transient. A protocol that tolerates the rapid dynamics of nodes and links could greatly improve the delivery rate of network packets. DTN presents an approach to address poor connectivity issues of certain mobile ad hoc networks [2]. An approach called *epidemic routing* has been proposed to deliver application data with high probability even when there is never a fully-connected path between source and destination, or when a network partition exists at the time a packet is originated [3].

Simulation tools are frequently used for wireless network research. Compared to experimentation using a hardware testbed that lacks flexibility, the software simulation provides an advantage to study new protocols or models in term of the cost and flexibility [4]. The open-source ns-2 simulator [5] has been widely employed in the academic research community. However, in response to a number of deficiencies, the ns-3 discrete event network simulator [6] has been developed as its successor, which ns-3 provides greater flexibility, evolvability, and modularity than ns-2. In addition, ns-3 supports heterogeneity including hybrid wired and wireless models. A set of MANET routing protocols have been implemented including OLSR (Optimised Link State Routing) [7], AODV (Ad Hoc On-Demand Distance Vector) [8], DSDV (Destination-Sequenced Distance Vector) [9], and DSR (dynamic source routing) [10] protocols. Of the four MANET routing implementations, DSDV [11] and DSR [12] have been implemented by our research group. DTNs (delay-tolerant network) have become an extremely

important research topic in recent years. However, there are currently no DTN routing protocols available in the ns-3 distribution. We provide a detailed implementation of the epidemic routing protocol in ns-3, which fills this gap and makes ns-3 a more comprehensive simulation tool for wireless network research.

In this paper, we present our ns-3 implementation of the epidemic routing protocol, and compare its performance with other existing MANET routing protocol models in ns-3. The remainder of this paper is arranged as follows: In Section 2, we present background and related work on epidemic routing and its implementations. Section 3 presents the implementation details of epidemic routing in ns-3. Section 4 evaluates the performance of our implementation of epidemic routing protocol and compare with ns-3 implementation results. A comparison of epidemic routing against several MANET routing protocols in ns-3 is presented in Section 5. Finally, we summarise our work and discuss future steps in Section 6.

2. BACKGROUND AND RELATED WORK

In this section, we present background about DTNs and epidemic routing, and briefly describe the MANET routing protocols that we use to compare with our epidemic routing protocol implemented in ns-3. Finally, we investigate some simulation tools that have been used in DTN research.

2.1 MANET Routing Protocols

MANET routing protocols can be classified into two categories based on their update mechanisms: proactive and reactive. Proactive routing protocols maintain updated routing information of all the nodes in the network by periodically distributing routing information among each other. The advantage is that the routes to any destination are ready to use when needed. However, forwarding tables grow with the size and node density of the network, rather than the number of routes actually needed. The overhead of flooding route advertisements to maintain convergence is a major drawback of proactive protocols. DSDV distance vector [9] and OLSR link state [7] are two well-studied proactive routing protocols.

Unlike proactive routing protocols, reactive routing protocols construct routes only when needed for data transmission. When a route to a new destination is required, the node initiates a route request and must wait until the route is discovered. There is no need to distribute routing information periodically or to maintain routes for all the nodes in the network. The disadvantage is the delay in finding routes to new destinations. DSR [13] and AODV [14] are two well-studied reactive routing protocols. DSR is an on-demand routing protocol based on the source routing concept. AODV is a distance vector routing protocol that operates only on demand.

2.2 Delay-Tolerant Networking

Unlike MANETs, DTNs are frequently partitioned such that end-to-end paths may not exist. In a mobile ad hoc network environment, dynamic and intermittent connectivity resulting from node mobility and range poses a significant challenge to the normal operation of the network. Network performance degrades due to the frequent partition of dynamic topologies. However, it is possible that there may be eventually a temporal path from the source to the destination by utilising the node mobility. A DTN architecture has been proposed for challenged network environments [2, 15]. DTNs provide communications for a wide range of networks that might have exceptionally poor connectivity. An approach called epidemic routing has been proposed to deliver packets under the situation where there are no available paths between source and destination at the moment of packet sending [3].

2.3 DTN Routing Simulation Tools

Epidemic routing has been implemented and analysed previously, specifically in ns-2 [5]. The implementation in this work is based on the previous ns-2 work. A Java based simulator called the ONE (Opportunistic Network Environment) has been used for research in DTNs including its variants such as OMNs (Opportunistic Mobile Networks) [16]. A comparison of various DTN routing protocols has been performed using a simulator developed by DTNRG (The Delay-Tolerant Networking Research Group) [17], in which the topology dynamics are assumed to be known in advance. A novel social-based forwarding algorithm called BUBBLE has been proposed that can utilise the social and network structural metrics to enhance delivery performance [18].

3. EPIDEMIC MODULE FOR ns-3

In this section, we explain our epidemic routing implementation in detail. Epidemic routing has been implemented and analysed in ns-2 [3], on which we base our ns-3 implementation. The epidemic routing protocol uses a controlled-flooding mechanism to minimise overhead in the network. The UML diagram is shown in Figure 1. This diagram shows the relationship between classes used by the epidemic routing protocol. There are several conceptual data structures that are important to support the epidemic operation. Here we discuss the essential entities. In Section 3.1, we show several user parameters that are added to control the behaviour of epidemic routing such as buffer length. A beacon mechanism is used to advertise node locations, which is discussed in detail in Section 3.2. The packets are exchanged between nodes as they come in range, discussed in Section 3.3. We discuss how all these epidemic modules interact and operate in Section 3.4. Every node maintains a buffer to store data packets, for which a detailed explanation is presented in Section 3.5.

3.1 Epidemic Parameters

There are several parameters used to initialise the epidemic routing protocol, as shown in Table 1. In this section, we present our suggestions for choosing a value for each parameter. First, the `HopCount` represents the maximum number of times a packet is flooded before being discarded. This value should be equal to or greater than the network diameter in order to avoid discarding a packet before it is delivered between the farthest communicating nodes. The `QueueLength` indicates the maximum number of packets that a queue can hold. To avoid discarding packets due to an excess of packets, this value needs to be greater than the number of distinct packets in the whole simulation. The `QueueEntryExpireTime` is used to set the maximum time a packet can live in the epidemic queues since it is generated. Choosing a value for this parameter depends on application requirement for latency. The `BeaconInterval` indicates the time in seconds after which a beacon packet is broadcast. This parameter should be selected based on data rate of

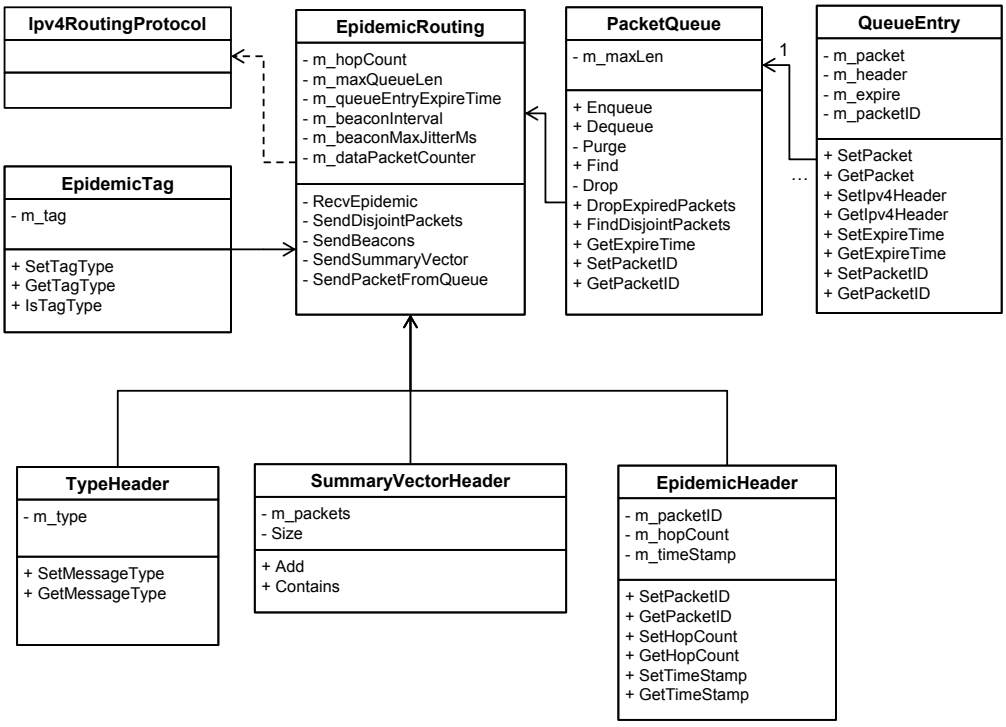

Figure 1: UML class diagram for epidemic routing protocol in ns-3

packet generation to get minimum delivery delay. For example, if one packet is generated every five seconds, it is recommended to set `BeaconInterval` to 5 seconds to exchange newly generated packets. In our simulations, the simulation time increases significantly as `BeaconInterval` decreases because more control packets are generated. The `BeaconRandomness` represents the upper bound of the uniform distribution of random time added to the beacon interval to avoid collisions, measured in milliseconds. When increasing the number of nodes and decreasing the `BeaconInterval`, the probability of collisions increases. To minimise the probability of collisions, `BeaconRandomness` should be set as an equivalent value to `BeaconInterval` to spread the beaconing uniformly throughout the simulation. The `HostRecentPeriod` is used to set the time in seconds for the period, in which hosts cannot re-exchange summary vectors. The main objective of this functionality is to eliminate redundant exchange summary vector sessions. This value should be greater than `BeaconInterval`.

3.2 Beacon Mechanism

The original implementation of epidemic routing, which was done in the ns-2 simulator, used a protocol named the Internet MANET Encapsulation Protocol (IMEP). IMEP provides the epidemic routing protocol with a notification service which informs the routing protocol when two nodes come in range of one another. However, IMEP is not implemented in the ns-3 network simulator as of the time of our epidemic implementation. Therefore, we have decided to add a beaconing mechanism as part of our epidemic routing implementation for ns-3. Our beaconing mechanism is basically the broadcasting of control packets bundled with information about the sender. When a node receives a beacon, it will know that it is in range of the node that transmitted the beacon packet. In our implementation, the user can specify how frequently a beacon is sent via the `BeaconInterval` parameter.

3.3 Summary Vector Exchange

In this section, we describe the *summary vector exchange* mechanism, which is how two nodes exchange their disjoint packets when they move in range of each other. The objective of the summary vector exchange is to avoid sending packets already present at the other node. First, we assume that there are two nodes approaching each other in the same area, labelled A and B. Once these nodes come into range of one another, they start the summary vector exchange session as shown in Figure 2. In order to control the summary vector exchange session, we use the concept of anti-entropy sessions [19], which states that the smaller node identifier must start the session. In this implementation, the rightmost two octets of a IPv4 address are used as a node identifier (ID). Assuming that node A has a smaller identifier, A sends its summary vector, denoted here as SV_A, to node B. After receiving SV_A, B compares the packet IDs presented in A's summary vector with the packet IDs located in its own buffer, and determines the disjoint set of packet. This set is then sent back to node A. Finally, node B sends its summary vector to node A, which also determines the disjoint set of packets and sends them back to node B. At this point both nodes have the same set of packets.

Table 1: Epidemic attributes and their default values

Attribute	Defaults	Summary
`HopCount`	64	Maximum number of times a packet will be flooded
`QueueLength`	64	Maximum number of packets that a queue can hold
`QueueEntryExpireTime`	100	Maximum time a packet can live in the epidemic queues since generated at the source
`HostRecentPeriod`	10	Time in seconds for host recent period, in which hosts can not re-exchange summary vectors
`BeaconInterval`	1	Time in seconds after which a beacon packet is broadcast
`BeaconRandomness`	1000	Upper bound of the uniform distribution random time added to avoid collisions, in milliseconds

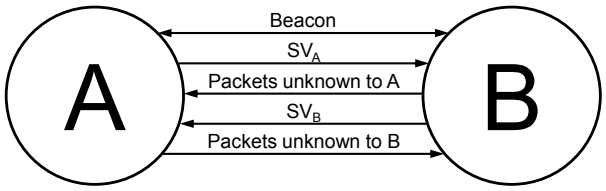

Figure 2: Summary vector exchange scheme

3.4 Epidemic Operation

In this section, we explain how epidemic routing functions operate. The epidemic routing class is inherited from Ipv4RoutingProtocol, which contains several functions for basic routing operations. It starts by initialising a `PacketQueue` to store buffered packets and other parameters discussed in Section 3.1. Using the `SendBeacons` function, each node advertises its presence to other nodes in the network. Once another node receives a beacon, it is processed using `RecvEpidemic` function to trigger the summary vector exchange process explained in Section 3.3. During this session, a summary vector containing the packet IDs is sent using `SendSummaryVector`. Once the summary vector is received, disjoint packet IDs are sent back using `SendDisjointPackets` function. Actual disjoint packets are forwarded using the `SendPacketFromQueue` function. After this process, both nodes should have identical packets in their buffers. If a packet reaches the destination, it will be delivered to the transport layer.

3.5 Epidemic Buffer

Each node maintains a `PacketQueue` that contains buffered packets. Each buffered packet is stored in a `QueueEntry` object that contains several elements: `packet`, `header`, `ucb`, `ecb`, `expire`, and `packetID`. The `packet` and `header` store the buffered packet and its header respectively. The `ucb` stores callback functions to forward and `ecb` raises an exception in case of errors. The `expire` indicates the expiration time of the packet while `packetID` indicates the global packet ID of the buffered packet. The `PacketQueue` class contains several functions to assist maintaining the epidemic queue. For example, the `PacketQueue::GetSummaryVector` function returns the summary vector of a current node's buffer. The `PacketQueue::FindDisjointPackets` function determines and constructs a summary vector that contains the disjoint packets between the given list and current buffer. Next, the `PacketQueue::DropExpiredPackets` function is used to drop expired packet in the current node's buffer.

3.6 Epidemic Header Format

The epidemic routing protocol uses three types of headers to implement its functionality: `TypeHeader`, `SummaryVectorHeader`, and `EpidemicHeader`. In this section, we present these types and their format.

3.6.1 TypeHeader

`TypeHeader` is an 8-bit header added to control packets to indicate their types once they arrive to the destination. They are used to control the summary vector exchange session. There are three types of control packets: `Beacon` packet type is used to advertise the presence of a node in the network. Once a beacon packet is received, `Reply` packet type starts the anti-entropy session. The node with smaller node ID will send a reply packet, which contains summary vector of all the packet IDs in its buffer. `Reply Back` packet type is used once a reply packet is received; the receiver determines the disjoint packets between its buffer and the received summary vector. Then, the receiver sends its disjoint packets to the sender. After that, the sender sends a `Reply Back` packet containing a summary vector of all the packet IDs in its buffer so the other host sends the disjoint packets as well.

3.6.2 SummaryVectorHeader

The `SummaryVectorHeader` is a variable size header that begins with a 32-bit length of the summary vector, denoted as n. Next, n 32-bit global packet IDs are inserted. This header is added to the `Reply` and `Reply Back` packets during summary vector exchange session. Since the UDP packet header is used to carry the summary vector, the length of the summary can not exceed 2^{16} bits. The summary vector packet is determined and generated using the `PacketQueue::GetSummaryVector` function. On the receiver side, the disjoint packet summary vector is determined using the `PacketQueue::FindDisjointPackets` function that returns an `SummaryVectorHeader` object. The format of `SummaryVectorHeader` is shown in Figure 3.

3.6.3 EpidemicHeader

This packet header is added to a data packet in the source node once received from the transport layer. It is removed in the receiver node before it is delivered to the transport layer. In our implementation, the `EpidemicHeader` has a fixed-length portion with a width of 128 bits, split across three fields: `packetID`, `hopCount`, and `timeStamp`. The 32-bit global packet identifier is stored in the `packetID` variable as the packet is created. The format of a *global packet ID* is a concatenation of 16-bit sender IP and a 16-bit sender data

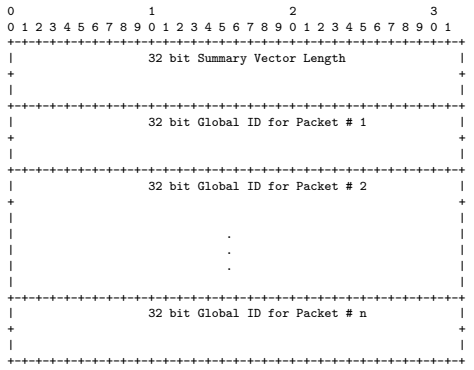

Figure 3: `SummaryVectorHeader` format

packet counter. We denote the packet ID *global packet ID* to distinguish from the ns-3 packet ID. A user-specified 32-bit hop count value is stored in the `hopCount`. It is a flood-control parameter used to restrict the number of hops the packet can travel before it is discarded. It is similar to the IP TTL field but with a higher size limit. A 64-bit time of creation value is stored in the `timeStamp` variable. This field is used to discard old packets with a time threshold limit set by the user. Both `packetID` and `timeStamp` do not change as the packet travels through the network, while `hopCount` is decremented by one after each forwarding step. The format of `EpidemicHeader` is shown in Figure 4.

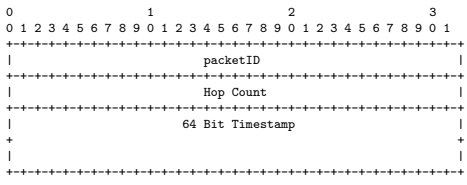

Figure 4: `EpidemicHeader` format

4. EPIDEMIC MODULE EVALUATION

To verify and validate our epidemic routing protocol implementation, first we run several unit test cases incorporated in the epidemic module of ns-3 and verify its basic functionality. Then, using several scenarios, we study packet delivery as a function of time delay while varying transmission range, hop count, and buffer size.

4.1 Simulation Setup

We configure simulation parameters as closely as possible to previous ns-2 studies [3] in order to have comparable results. We perform the simulations over an area of 1500 $\times$ 300 m^2. All the simulations are averaged over 100 runs with each simulation running for 200,000 s. Simulations are performed with 50 nodes, with a subset of 45 nodes selected as sources and sinks. Each of the 45 nodes sends one packet to the other 44 nodes with a total of 1980 packets at 10 s simulation time. We perform simulations with a packet size of 1024 bytes. We use the ns-3 `On-Off` application to generate CBR (constant bit rate) traffic. The 802.11b MAC is the link layer over the range propagation loss model to limit the transmission ranges of nodes. The transmission range of the nodes is set as 100 m for evaluation. The mobility model used is steady-state random waypoint [20] with random velocities from 0.01 – 20.0 m/s. A summary of all simulation parameters is shown in Table 2. Furthermore, a summary of the Epidemic routing protocol parameters is shown in Table 3.

Table 2: Simulation parameters

Simulation Parameters	Value
Simulation area	1500 m × 300 m
Number of runs	100
Warmup time	10 s
Total simulation time	200,000 s
Mobility model	Random waypoint
Node speed	0 – 20 m/s
Packet size	1024 bytes
Number of packets	1980 packets/simulation
Link layer	wifib-11mbs
Propagation loss model	Range

Table 3: Epidemic evaluation parameters

Parameter	Values
HopCount	10
QueueLength	2,000
QueueEntryExpireTime [s]	200,000
BeaconInterval [s]	5,000
BeaconRandomness [s]	5
HostRecentPeriod [s]	10

4.2 Simulation Analysis

Based on the technical report [3] that describes the implementation of epidemic routing in ns-2, there are three scenarios for evaluating epidemic routing protocol performance. In the first scenario, the transmission range is varied from 10 m to 250 m. In the second scenario, the number of hops is varied from 1 hop to 8 hops. In the third scenario, the buffer length is varied from 10 packets to 2000 packets. In all scenarios, the percentage of packets delivered is plotted as a function of packet delivery latency. Other routing protocol parameters are set to match the original report to yield comparable results.

4.2.1 Varying Transmission Range

In this scenario, the transmission range is varied with five different values: {10, 25, 50, 100, 250} meters. With all of these transmission range values, we get *eventually*, 100% packet delivery. However, as transmission range decreases, the packet delivery latency increases. This is because when the transmission range is small (e.g. 10 m), the connectivity becomes highly intermittent, which causes the network to have a higher number of disconnected components consisting of a small number of nodes. On the other hand, as the transmission range increases, components merge together. As a result, the packet delivery latency is much shorter for 250 m than 10 m, as shown in Figure 5. We note that in the original ns-2 implementation, the technical report shows that the packet delivery is 89.9% for 10 m transmission range while in our implementation the packet delivery reaches 100% as expected with high delay. We believe that the original pa-

per results does not show 100% because there were some artifacts in ns-2 implementation.

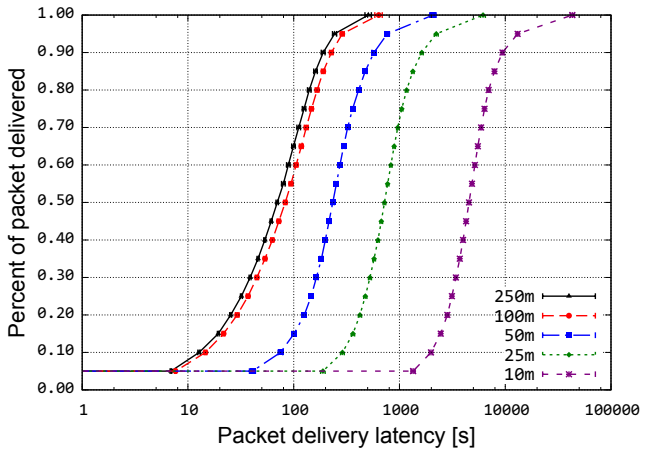

Figure 5: CDF for varying transmission range

4.2.2 *Varying Hop Count*

In this scenario, the hop count is varied with five different values: {1, 2, 3, 4, 8} hops. Similar to the varying transmission range, packet delivery is 100% for all hop count values. However, as hop count decreases, the packet delivery latency increases as shown in Figure 6. This is because packets with a small hop count get dropped faster than packets with a higher hop count. For example, packets with a 1-hop count value do not get delivered until the source and sink nodes come in range of each other, which takes more time since nodes are moving around randomly. However, with an 8-hop count value, packets can be spread epidemically to reach the sink faster while the they can travel through up to 8 intermediate hops. Again, we note that in the original ns-2 implementation, the technical report shows that the packet delivery rate is 80–90% for 1 and 2 hop count values range while in our implementation the packet delivery is 100% as expected with high delay.

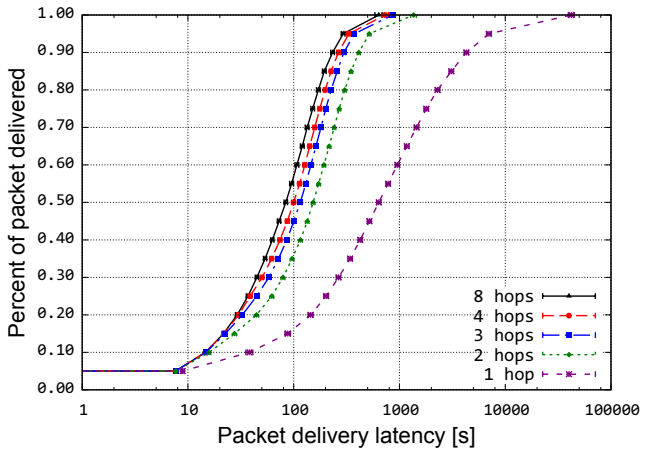

Figure 6: CDF for varying hop count

4.2.3 *Varying Buffer Length*

In this scenario, the buffer length is varied with eight different values: {10, 20, 50, 100, 200, 500, 1000, 2000} packets. The results are shown in Figure 7. In the scenario with 2000 buffer length, the packet delivery is 100%. This is because there are 1980 packets in each simulation, the length 2000 emulates an infinite buffer. However, for the other values of buffer lengths, the packet delivery decreases proportionally with buffer length. This is because packets are dropped when node buffers reach their capacity. This shows that the buffer capacity is a critical parameter and it should be set large enough to contain all packet expected in the network to obtain 100% packet delivery in this scenario.

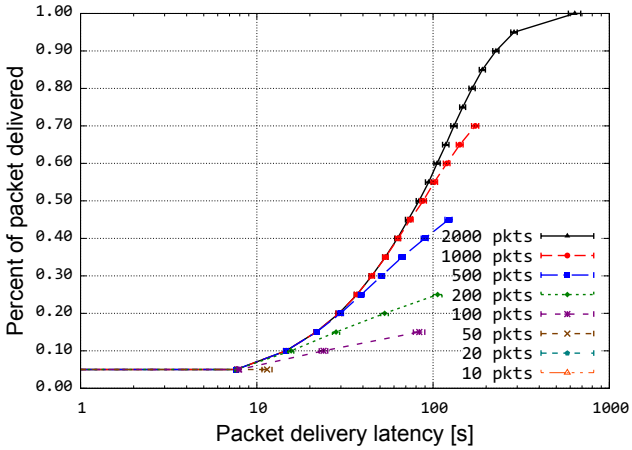

Figure 7: CDF for varying buffer size

5. MANET COMPARATIVE EVALUATION

We further carry out comparison of the epidemic routing protocol with four major MANET routing protocols: AODV, DSDV, DSR, and OLSR. The metrics for comparison are PDR (packet delivery ratio), end-to-end delay, and routing overhead. The simulation environment is relatively sparse with varying number of nodes.

5.1 Simulation Setup

We perform the simulations over an area of 1500×300 m^2. All the simulations are averaged over 10 runs with each simulation running for 2000 s. Simulations are performed with five source-sink traffic pairs with the varying number of nodes. We perform simulations with a packet size of 64 bytes to exclude the potential network congestion caused by large packets and highlight the performance of routing protocols [21]. All the nodes are configured to send 1 packet/s. Using this lower packet rate, we can correctly evaluate the performance of the routing protocols. We use the ns-3 `OnOff` application to generate CBR (constant bit rate) traffic. The 802.11b MAC is the link layer over the range propagation loss model to limit the transmission ranges of nodes. Previous routing protocol performance study used a combination of free space propagation model and a two-ray ground reflection model [21]. The transmission range of the nodes is set at 100 m for evaluation. The mobility model

Table 4: Comparison parameters

Comparison Parameters	Value
Simulation area	1500 m × 300 m
Number of runs	10
Warmup time	100 s
Total simulation time	2000 s
Mobility model	Steady-state RWP
Node speed	20 m/s
Packet size	64 bytes
Number of packets	1000 packets/simulation
Propagation loss model	Range

used is steady-state random waypoint[1] with node velocity at 20.0 m/s. The pause time is zero meaning that the nodes are constantly moving. The simulation parameters are shown in Table 4.

5.2 Simulation Results

The epidemic routing protocol demonstrates its effectiveness in delivering 100% data packets in a sparse wireless environment[2] as shown in Figure 8. We observe that as the number of nodes increases, the packet delivery ratio for the other routing protocols increases. We notice when the number of nodes is 30, there is a PDR bump for all the four MANET protocols. The reason we believe is that 30 nodes in this simulation area realises a "sweet" node density. As the node density continues to increase, the PDR drops as the protocols are forced to form longer routes, which are easier to break with node movements.

We configure the queue length and queue expiration time for all the protocols to match epidemic. The queue length is set as 1000 which is effectively infinite[3] with the total simulation time as expiration time. OLSR does not have a data queue configuration, so we leave it unchanged. Epidemic has a beaconing mechanism set as 5 s interval.

This routing overhead increases with the increasing number of nodes as shown in Figure 9. The overhead is in kb/s. Epidemic has a beaconing mechanism and as the node number increases, the total number of beaconing packets increases exponentially. DSDV and OLSR proactively send out control packets to find routes, so the route overhead increases with the number of nodes. The DSR and AODV increase of overhead is minimal since they only react to route changes; with 20 m/s node velocity, this is very limited.

The packet delay for epidemic is high compared to the other MANET protocols as shown in the Figure 10. The delay is in ms and the y-axis is log scale. The is expected since epidemic queues data packet guarantees eventual packet delivery, but the node delay could be extremely high in a sparse routing environment. As the number of nodes increases, the overhead for epidemic decreases; it is a controlled broadcast protocol, as the number of nodes increase, the chance for node contact is higher. As the number of nodes increases from 10 to 50, the delay drops about 80%. In contrast for the other routing protocols, the delay increases with the number of nodes. This is because only the delivered packets are counted as delay, and a significant percentage of packets are dropped. For the delivered packets, the hop-count for the routes increases as the number of nodes increases, hence the increase of delay.

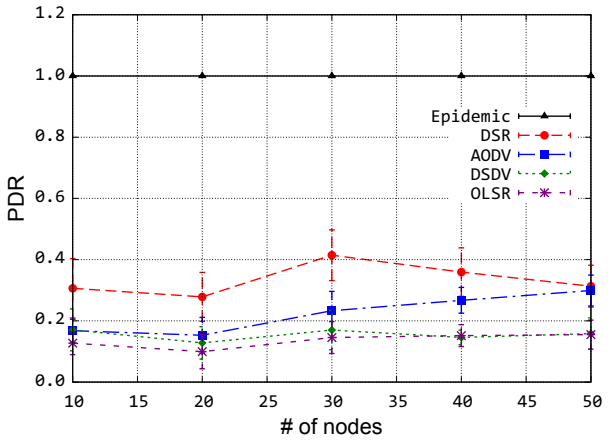

Figure 8: PDR for varying number of nodes

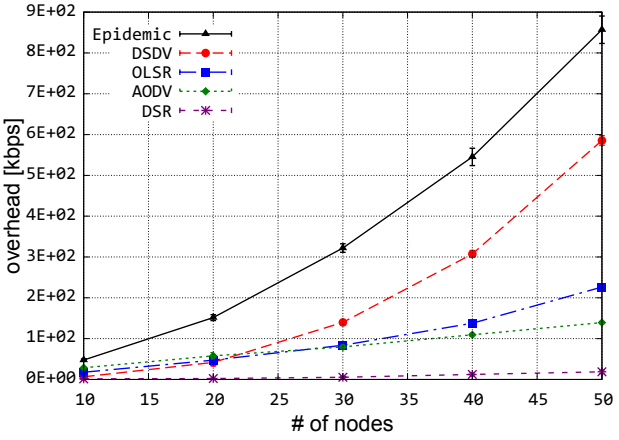

Figure 9: Overhead with varying number of nodes

6. CONCLUSIONS

Several routing protocols have been proposed for MANET environments to improve overall performance in terms of packet delivery and end-to-end delay. In this paper, we presented our implementation of the epidemic routing protocol in ns-3. We provide explanations about each complement of our implementation and how they interact with each other. We have added several user parameters beyond the original ns-2 implementation to modify the routing behaviour and presented guideline for choosing each parameter. We analysed our epidemic implementation with varying transmission range, number of hops, and buffer size. Our results indicate that epidemic routing provides an eventual 100% packet delivery for different transmission ranges and hop count values as long as the buffer size has enough space for all packets. However, as buffer size decreases, the packet delivery decreases proportionally. Furthermore, we compare

[1]This model eliminates the initial discrepancy of the RWP model and uses a stationary distribution.
[2]100 m transmission range in an area of 1500 × 300 m^2 with 10 nodes
[3]200 seconds at one packet/s for five source-sink traffic pairs generates 1000 packets

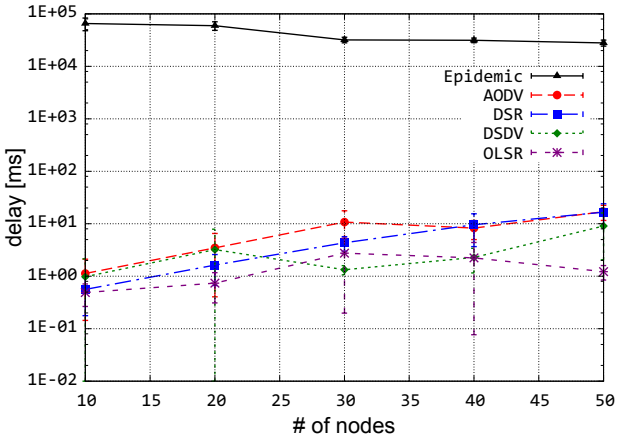

Figure 10: Packet delay for varying number of nodes

this epidemic routing protocol to other baseline MANET protocols in a DTN environment. The results show that epidemic routing provides higher packet delivery rate with high delay compared to other baseline MANET protocols. Furthermore, our epidemic routing implementation is under review for going to the main release distribution[4].

7. ACKNOWLEDGMENTS

We would like to acknowledge Peter Barnes for reviewing our routing protocol and helping in fixing multiple software bugs. Moreover, we would like to acknowledge the assistance of the members of the ResiliNets research group for their advice and suggestions which helped us with this implementation. We would also like to thank Tom Henderson and the ns-3 development team for their responsiveness to issues with ns-3 platform.

8. REFERENCES

[1] I. Chlamtac, M. Conti, and J. J.-N. Liu, "Mobile ad hoc networking: imperatives and challenges," *Ad Hoc Networks*, vol. 1, no. 1, pp. 13–64, 2003.

[2] K. Fall, "A Delay-Tolerant Network Architecture for Challenged Internets," in *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, New York, NY, USA, pp. 27–34, ACM, 2003.

[3] A. Vahdat and D. Becker, "Epidemic routing for partially-connected ad hoc networks," Technical Report CS-200006, Duke University, April 2000.

[4] S. Kurkowski, T. Camp, and M. Colagrosso, "MANET simulation studies: the incredibles," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 9, no. 4, pp. 50–61, 2005.

[5] "The network simulator: ns-2." `http://www.isi.edu/nsnam/ns`, December 2007.

[6] "The ns-3 network simulator." `http://www.nsnam.org`, July 2009.

[7] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)." RFC 3626 (Experimental), October 2003.

[4]Source code is at https://codereview.appspot.com/13831049

[8] C. Perkins and E. Royer, "Ad-hoc On-demand Distance Vector Routing," in *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, pp. 90–100, February 1999.

[9] C. E. Perkins and P. Bhagwat, "Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers," in *ACM SIGCOMM*, London, UK, pp. 234–244, 1994.

[10] D. Johnson, Y. Hu, and D. Maltz, "The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4." RFC 4728 (Experimental), February 2007.

[11] H. Narra, Y. Cheng, E. K. Çetinkaya, J. P. Rohrer, and J. P. Sterbenz, "Destination-sequenced distance vector (DSDV) routing protocol implementation in ns-3," in *Proceedings of the ICST SIMUTools Workshop on ns-3 (WNS3)*, Barcelona, Spain, pp. 439–446, March 2011.

[12] Y. Cheng, E. K. Çetinkaya, and J. P. Sterbenz, "Dynamic source routing (DSR) protocol implementation in ns-3," in *Proceedings of the ICST SIMUTools Workshop on ns-3 (WNS3)*, Sirmione, Italy, pp. 367–374, March 2012.

[13] D. B. Johnson, D. A. Maltz, and J. Broch, "DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks," in *Ad Hoc Networking* (C. E. Perkins, ed.), ch. 5, pp. 139–172, Boston, MA, USA: Addison-Wesley, 2001.

[14] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing." RFC 3561 (Experimental), July 2003.

[15] J. P. Sterbenz, R. Krishnan, R. R. Hain, A. W. Jackson, D. Levin, R. Ramanathan, and J. Zao, "Survivable Mobile Wireless Networks: Issues, Challenges, and Research Directions," in *Proceedings of the 1st ACM Workshop on Wireless Security (WiSe)*, Atlanta, GA, pp. 31–40, September 2002.

[16] "The ONE simulator." `http://www.netlab.tkk.fi/tutkimus/dtn/theone`.

[17] "DTN research group." `https://sites.google.com/site/dtnresgroup`.

[18] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: Social-based forwarding in delay-tolerant networks," *IEEE Transactions on Mobile Computing*, vol. 10, pp. 1576–1589, November 2011.

[19] D. B. Terry, M. M. Theimer, K. Petersen, A. J. Demers, M. J. Spreitzer, and C. H. Hauser, "Managing update conflicts in bayou, a weakly connected replicated storage system," in *Proceedings of the Fifteenth ACM Symposium on Operating Systems Principles*, SOSP '95, New York, NY, USA, pp. 172–182, ACM, 1995.

[20] W. Navidi and T. Camp, "Stationary distributions for the random waypoint mobility model," *Mobile Computing, IEEE Transactions on*, vol. 3, no. 1, pp. 99–108, 2004.

[21] J. Broch, D. Maltz, D. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," *Proceedings of the 4th annual ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 85–97, October 1998.