# e-ONE:Enhanced ONE for Simulating Challenged Network Scenarios

Sujoy Saha[a], Rohit Verma[b], Somir Saika[a], Partha Sarathi Paul[b], Subrata Nandi[b]
[a] Department of Computer Applications, National Institute of Technology, Durgapur 713209, India
Email: {sujoy.ju, somirsaikia}@gmail.com
[b] Department of Computer Science & Engineering, National Institute of Technology, Durgapur 713209, India
Email: {rohitverma.kgp, mtc0113, subrata.nandi}@gmail.com

***Abstract*— Delay Tolerant Network (DTN) empowers sparse mobile ad-hoc networks and other challenged network environments, such as interplanetary communication network or deep sea communication network, where traditional networking protocols either fail to work completely or do not work well. The Opportunistic Networking Environment (ONE) Simulator has gained considerable popularity as an efficient tool for validating and analysing DTN routing and application protocols. It provides options for creating different mobility models and routing strategies as per the users' requirements. Nowadays, challenged networks such as rural internet connection, social networks, post-disaster communication systems, etc. use DTN along with some hybrid infrastructure networks. Incorporating such real life network systems in ONE needs extensive modification of the same. In this paper, we present the enhanced ONE (e-ONE) simulator as an extension of ONE to facilitate simulation of challenged networks and describe the enhancements we have added to the ONE. As a case study, we consider a challenged network, which we call a latency aware 4-tier planned hybrid architecture designed for post-disaster management. We describe, in detail, how this enhanced version of the ONE simulator is useful in analysis and evaluation of the scenario considered.**

***Index Terms*— e-ONE Simulator, Delay Tolerant Network, ONE Simulator, Ad Hoc Hybrid Network, Post Disaster Management**

## I. Introduction

Delay Tolerant Networking (DTN) aims at partially supporting an architecture with heterogeneous networks where there is a lack of continuous network connectivity. The adaptation of the Interplanetary Internet (IPN) ideas to terrestrial networks led to the birth of Delay Tolerant Networking [1]. Since then, there has been a considerable amount of research in this field by several groups around the globe. The concept has been well utilized for Interplanetary Communication and Deep Sea Communication. Current Internet protocols (i.e., the TCP/IP protocol stack) suffer and sometimes fail under such testing conditions; so we require a different category of network (routing) and transport layer protocols, which are tailor-made for challenged environments.

In recent times, researchers have focused on the use of DTN in challenged networks, where nodes is sparse, which results in an intermittent connectivity [2][3][4], and where only low end devices are available. Such network

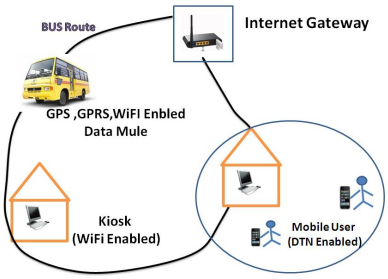



Figure 1. Low Cost Rural Internet Communication System

systems use, in addition to DTN, infrastructure networks comprising cellular networks, satellite communication systems, mesh networks etc. Examples include (i) heterogeneous architecture to provide low cost Internet service [5] and reliable connectivity to rural kiosks, using buses and cars as mechanical back-haul, [Figure 1] to ferry data to and from kiosks; (ii) hybrid architecture with different available technologies for post-disaster communication system [Figure 2] [6][7]; (iii) Twitter applications used in disaster mode relies on opportunistic communication and epidemic routing of tweets from phone to phone [8]. The tweets are transferred to the outside world when some pockets of network are eventually detected by the smart phones.

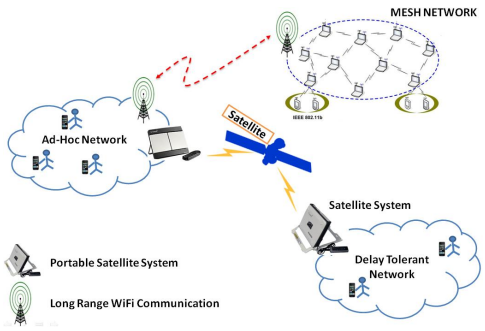



Figure 2. Hybrid Network for Post Disaster Management

All the above challenged network scenarios use hybrid network resources such as opportunistic/delay tolerant network at bottommost tier, some infrastructure based network, either through data mules (e.g bus,


doi:10.4304/jnw.9.12.3290-3304

ambulance, boat,UAV etc.) as mechanical back-haul or through MESH networks at middle tier, and long range WiFi/WiMax or satellite phone communication at topmost tier.

### A. Objective

The objective in this paper is to identify what needs to be done on the ONE simulator, which is developed explicitly for DTN protocol evaluation, so that it can be used to simulate, analyse and evaluate above mentioned real-world challenged network scenarios. Furthermore, the identified changes need to be incorporated into the existing ONE simulator in order to extend it to the enhanced ONE simulator.

Simulation tools for DTN protocol evaluation are not available in abundance. The only simulators available for evaluation of DTN are the ONE Simulator [9] and DTNSim2 [10], the latter being based on the DTNSim. All these simulators have been developed in Java using object oriented characteristics. The Opportunistic Network Simulator (ONE) is a widely used simulator for DTN protocol evaluation. It has numerous in-built features to enhance its expertise. It supports (i) different routing algorithms such as Epidemic routing [11], Prophet routing [12], Spray and Wait routing [13], Spray and Focus routing [14], MaxProp routing [15] etc.; (ii) different mobility models such as Random Waypoint model, Random Walk model, Shortest Path Map Based model and so on. In addition to these, there is a provision for applying real-life mobility traces in the form of text file called external movement, which enables the users in ONE to use practical movement scenarios in simulation instead of artificial random simulation. (iii) Nice visualization tool and GUI is available to set the simulation parameters (dynamically in some cases) and to observe the simulation progress, with user-friendly graphical representation. (iv) It is an open source package, providing the flexibility to modify different modules according to specific requirements. (v) Every new release of ONE adds some new routing algorithms and mobility models; several new features have been added by the community working with ONE. (vi) ONE also provides several metrics to analyse simulations, like latency, packet delivery probability, overhead, etc., which are echoed in the form of report files. Map of any location can also be used as the play field for simulations and the the simulation can be designed accordingly. (vii) Extensive documentation of the software makes ONE developer-friendly.

### B. Motivation

Although it is quite an effective simulator for DTNs, it still has a lot of ground to cover before being apt for the real world challenged network scenarios in general. It requires several major modifications to different modules in order to work with more realistic scenarios. The mobility of nodes needs to be modified in order to match with the real world requirements, such as group mobility, different types of path based movements, etc. There is a lack of several heterogeneous/smart interfaces, which need to be incorporated for further enhancement, such as the satellite phone interface and other infrastructure supporting nodes like long range WiFi, Mesh etc. These are essential for ONE to be compatible with a real-world challenged network system. Heterogeneous communication resources may yield a hybrid type of network in order to achieve better performance. Moreover, such enhancement would require inclusion of hybrid application-specific routing strategies, since different resources use different technologies to work.

In section 2, we have described the basic model of the ONE simulator and how it has been modified in the e-ONE. Also, we explain how the simulator has been customized as outlined. Section 3 presents a case study of a hybrid network and how e-ONE has been instrumental in analysing, evaluating and shaping it. In section 4, we have concluded the paper with directions for future research.

## II. Customization of ONE simulator

Incorporating real world hybrid architecture in the simulator requires customization of most of the modules of ONE. Here, we discuss the major modifications incorporated in the simulator to develop e-ONE. These modifications include modification of the Cluster Movement Model, restricting the epidemic routing algorithm, making the nodes more intelligent and developing new modules for special types of communication. But a hybrid network would require more realistic mobility models, along with certain modifications in the routing strategies. Above all, it would require other types of nodes which operate on the principle of infrastructure based network. These modifications have been shown in Figure 3, which depicts a model of e-ONE simulator.

Novelty of e-ONE over ONE in terms of challenged network applications can be listed as follows:

- Mobility during and after any disaster is quite different from that in a normal scenario. Broken paths, destroyed bridges, scarcity of infrastructure networks restricts the movement of rescue personnels. In Mobility Model subsection, we introduced Postoffice Cluster Movement model that may fit in this scenario.
- Routing strategies for DTNs is different from that of infrastructure-based networks. But a hybrid network uses both types of network at different tiers. There are multiple categories of nodes existent in the network, and a node has to decide which node it will forward its packets to. The Routing Strategy subsection deals with this problem.
- At some tier we may use NLOS devices, which are layer 2 devices, for which it cannot operate in an interconnection which contain loops. So one needs

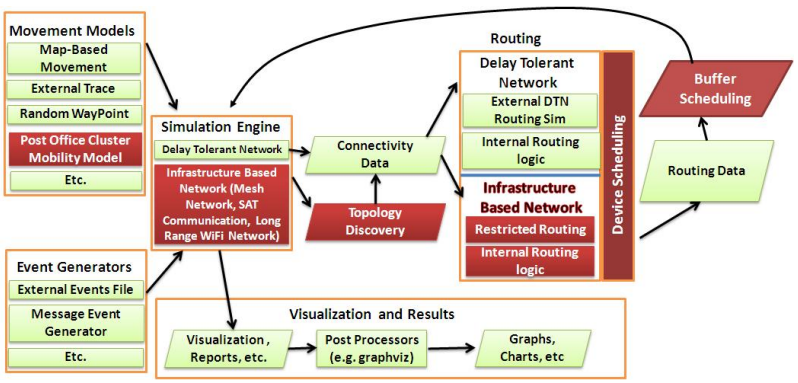



Figure 3. Existing ONE modules (in green) & added e-ONE modules (in red)

to implement a spanning tree algorithm, so that these devices can function.

- Since different types of networking devices are used, there exists the question of priorities for different nodes. In Device Scheduling subsection, we highlighted these priority scheduling of versatile types of networking devices used in our hybrid network.
- DTN nodes are accustomed in dealing with multiple copies of the same packet, which infrastructure-based networks are not. So suitable buffer scheduling is required for this.
- Depending on availability we may use sophisticated devices like satellite phones. So new interface modules needs to be implemented for such devices.

The above are our main thrust in upgrading ONE to e-ONE that may be used for simulating hybrid network models in challenged network scenarios.

### A. Mobility Model

The ONE simulator has provided us with a wide variety of mobility models to choose from. But none of them exactly matches with post-disaster situation movements. To bridge the gap, we have incorporated *post office cluster movement model (PCM)* as a modification of the Cluster Movement Model [16], and *poisson post office cluster movement model (PPCM)* as a further modification to post office cluster movement model. Algorithm 1 describes PCM and algorithm 2 describes PPCM in details. Both the strategies restricts node to a cluster with a specific range. There can be different clusters of varying range, each having a dedicated set of nodes (DTN nodes) whose movements are restricted to the cluster. This scenario has been shown in Figure 4.1. In either of the model, a specific location within the cluster is marked as the location of dropbox, to which point the nodes within the cluster visit regularly. The dropbox locations usually symbolizes the shelter points in a disaster-affected region, or location of rural kiosks in rural internet scenario.

*1) Post office cluster movement model:* In this movement model, each node inside the cluster move following a random waypoint movement strategy within the cluster, but it visits the location of dropbox after a fixed number of waypoints (Referred in Algorithm 1), the fixed number being chosen as a random integer between two fixed parameters $h_1$ and $h_2$ of the movement model. Figure 4.2 shows this scenario. This approach uses all the attributes and functions of the *ClusterMovementModel* class, with only a slight modification in the *randomCoord* function and a few aiding variables. The new *randomCoord* function keeps track of the number of hops by the host, and automatically sets as its next waypoint (Step 3) the location of the dropbox, when the maximum hop count is reached (Step 2).

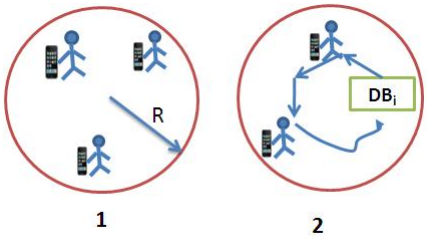



Figure 4. (4.1) Cluster Mobility Model (4.2) Post offfice Cluster Mobility Model

**Algorithm 1: PostOfficeClusterMovement**

Step 1: set random number of hops from the integer range [h1 , h2 ], between two consecutive visits within range R.
Step 2:**if** *(number of hops is equal to maximum hops)* **then**
  Step2.1: Bring the node back to center
**end**
Step 3: **else**
  Step3.1: Set next random coordinate
  Step3.2: Increase the number of hops by 1
  Step3.3: **return** *new coordinate*
**end**

*2) Poisson Post office cluster movement model:* In this movement model, inter-arrival times between two consec-

utive visits of a DTN node to the nearest Dropbox follows an exponential distribution. This is a one-parameter movement model in which the parameter $\lambda$ represents the average inter-arrival time between two consecutive visits (i.e for the exponentially-distributed time gaps) to the dropbox. After generating the next inter-arrival time gap, $T_1$ say, randomly following the corresponding exponential distribution with parameter $\lambda$, Algorithm 2 finds one random waypoint (Say $P_1$), calculate time from DB to $P_1$ plus time from $P_1$ to dropbox (called roundtrip time), and check whether $P_1$ is reachable or not within $T_1$. If $P_1$ is found to be reachable, then the node moves to $P_1$, generate next random waypoint, $P_2$ say (Step 9.1, 9.2 and 9.4) and check the time from dropbox to $P_1$ plus time from $P_1$ to $P_2$ plus time from $P_2$ to dropbox is within $T_1$ or not. If reachable, then the node moves to $P_2$ and generate the next waypoint $P_3$, and so on. If one randomly generated waypoint $P_i$ is found unreachable, then the corresponding point is regenerated and the same process is repeated. When the remaining time for returning the dropbox is too less (less than a threshold value $T_{Th}$, say) then it simply selects the dropbox as its next waypoint, returns the dropbox, and start the next iteration by generating the next inter-arrival time $T_2$, say.

**Algorithm 2: PoissonPostOfficeClusterMovement**

Step 1: Generate Next Poisson Number $T_{poisson}$.
Step 2: P $\leftarrow$ $P_{DB}$ ; //Intial location of DTN node; Start from nearest DB
Step 3: Total $\leftarrow$ $Total_1 \leftarrow 0$ ;
Step 4: Generate new point $P_1$ , new velocity $V_1$ , new pause time $T_{pause}$ .
//Time taken form point P to $P_1$
Step 5: $T_{P,P_1}$ $\leftarrow$ Distance$(P, P_1)/V_1$ ;
Step 6: $T_{P_1,P_{DB}}$ $\leftarrow$ Distance$(P_1, P_{DB})/V_{Max}$ ;
Step 7: $Total_1 \leftarrow Total_1$ + $T_{P,P_1}$ + Tpause ;
Step 8: Total $\leftarrow$ $Total_1$ + $T_{P_1,P_{DB}}$ ;
Step 9: **if** *(Total* $< T_{poisson}$*)* **then**
  Step 9.1: Move to $P_1$ with velocity $V_1$ and wait for pause time $T_{pause}$ ;
  Step 9.2: P $\leftarrow$ $P_1$ ;
  Step 9.3: **if** *(Total* + $T_{Th} \geq T_{poisson}$*)* **then**
    Step 9.3.1: $V_{next}$ $\leftarrow$ Distance($P_1$,$P_{DB}$)/ ($T_{poisson}$ - $Total_1$);
    Step 9.3.2: Return $P_{DB}$ with speed $V_{next}$ and wait for some Pause Time;
    Step 9.3.3: Go to Step 1;
  Step 9.4: **else**
    Go to Step 4 ;

### B. Routing Strategy

We have devised a strategy to restrict flooding in the epidemic routing algorithm. The basic algorithm ensures that the message is flooded to all nodes in range of the host node. In our strategy, message is not relayed to those nodes which have already received the message. While this requires additional checks to determine if a node in range has already received a particular message, this ensures that flooding is limited.

Furthermore, rather than treating the entire network like a set of homogeneous nodes, we have divided the network into several entities. Messages from a particular entity are only allowed to be relayed to a sub-set of all other entities, to make communication more meaningful. For example, a message for the head office in a local office would only be relayed to someone having access to the head office and not to someone in the local office.

The desired constraints were met by modifying the *startTransfer* function and adding the boolean function *shouldSendMessage* for each stage, which checked the above constraints and returned true or false as per the algorithm.

The *shouldSendMessage* function is application-specific, and hence the modified routing strategy would depend on the context in which the user is using the simulator. As we have used the *PostOfficeClusterMovement*, let us take an example of a set of clusters, with a set of carrier nodes, denoted $CN$, between them. Let there be some control points in each cluster, known as $CP$, where the information dropbox(DB)/Postbox is placed. So the CNs would move between these clusters to collect and relay inter-cluster messages from the DB of the cluster. In this scenario, the DB must only relay a message to to specific $CN$ which is moving towards the destination cluster (Step 2). If CNs are in the range of contact then one CN deliver the message to that CN which is towards the destination (Step 3). Algorithm 3 depicts the pseudo-code for this scenario.

### C. Topology Discovery for Infrastructure Nodes with Fault detection and Recovery

As Long Range WiFi Communication(LWC) devices are layer - 2 devices (TCP/IP Protocol Stack), they lack in intelligence of alternate route discovery and thus existence of loops in the network may hang the whole communication system. To avoid such unwanted behaviour, a spanning tree formation is mandatory in the network.

Set of LWC devices forms the vertex set of the LWC graph (Referred in Algorithm 4). The vertices in the graph is connected by an edge if the corresponding LWCs are in the range. Vertex (S) is predefined starting vertex.

Initially, we start with an empty set of processed nodes $Processed$ and an empty queue $Q$.

Vertex $S$ will be added to $Processed$ (Step 2) and in queue $Q$. Similar iterations will be carried out on other vertices until $Q$ is empty. In each iteration, we delete an element X (Step 3.1) from $Q$ and add all the nodes Y adjacent to X which has not been inserted to $Q$ (Step 3.2), edge (X,Y) will also be added to the output graph H. When $Q$ is empty, acyclic network of LWC devices is found in H, which is suitable for our purpose (Step 3).

**Algorithm 3: Routing Strategy**

**Input:** *Message m, Node Receiver_Node*
Step 1: Here, ***CN*** is the Carrier Node which moves among some Dropboxes (***DBs***); m is the message to be relayed currently in possession of host_node, and Receiver_Node is the node with which the host_node comes in contact with.
Step 2: **if** *(host_node is DB and Receiver_Node is CN)* **then**
  Step 2.1: **if** *(CN confirms DB to visits desired destination)* **then**
    Step 2.1.1: DB Transmit Packet to that particular CN.
  **end**
  Step2.2**else**
    Step 2.2.1: No Transmission of Packet
  **end**
**end**
Step 3: **else if** *(host_node and Receiver_Node both are CN type)* **then**
  Step3.1: **if** *(Receiver_Node is bounded towards the desired destination)* **then**
    Step3.1.1: Transmit Packet to that particular CN.
  **end**
  Step3.2: **else**
    Step3.2.2: No Transmission of Packet
  **end**
**end**

**Algorithm 4: Topology Discovery Algorithm**

**Input:** *Graph G(V,E) the graph network formed by the LWC devices where the Vertices are the LWC Devices and Edges represent devices in range, S : a Starting Station*
**Step 1 :** Processed = $\phi$ // a set of nodes which have been processed
  Q // an empty queue
  H // an empty graph
**Step 2 :** Add S to Processed
  Q.insert(S)
**Step 3 : while** *( Q is not empty )* **do**
  **Step 3.1 :** X = Q.delete()
  **Step 3.2 : for** *(all nodes Y which is Adjacent to X and not in Processed)* **do**
    **Step 3.2.1 :** add Y to Processed and Q.enqueue(Y)
    **Step 3.2.2 :** insert edge (X,Y) to graph H
  **end**
**end**
**Step 5:** Output: Graph H : the acyclic network of LWC devices (tree network)

The nodes periodically check the status of the nodes in their list with the help of *beacon* messages. If a node is found to be *down* (Step 1 of Algorithm 5), it is removed from the host's list and added to a temporary list. This

**Algorithm 5: Fault Detection & Recovery**

**Input:** *Connection C, Graph G*
*Connection C is connection between two LWCs and Graph G(V,E) the graph network formed by the LWC Devices and Edges represent devices in range*
Step 1: **if** *(Connection is Down in between two LWCs)* **then**
  Step 1.1: X=get first host of connection C
  Step 1.2: Y=get another host of connection C
  Step 1.3: **if** *(there is an edge between X & Y in Graph G)* **then**
    Step 1.3.1: Delete edge (X,Y) from graph G
    Step 1.3.2: Call Topology Discovery Algorithm on this new updated Graph G
  **end**
**end**
Step 2: **else**
  Step 2.1: **if** *(Connection* $< X, Y >$ *is up Between two LWCs)* **then**
    Step 2.1.1: **if** *(there is no edge between X & Y in Graph G)* **then**
      Step 2.1.1: Insert edge (X,Y) to graph G
      Step 2.1.2: Call Topology Discovery Algorithm on this new updated Graph G
    **end**
  **end**
**end**

list stores the node till it again becomes *active* (Step 2), which is again detected with the help of *beacon* messages. In the meantime, a new topology is followed for the nodes which are given by the algorithms 4 and 5.

### *D. Device Scheduling*

When two types of devices are connected to a common device, a priority value for each type of devices is set by common device and higher priority device should be served first by common device.

**Algorithm 6: Device Scheduling: Procedure start-Transfer(Node A , Node X )**

L = List of all devices connected to A
**for** *(all i in L)* **do**
  **if** *(priority( i ) > priority( X ))* **then**
    return false;
  **end**
**end**
return true;

The routine *Device Scheduling* finds all connected node with device (Referred in Algorithm 6) A based on priority value and selects the highest priority device for packet transmission.

### *E. Buffer Scheduling*

Buffer Scheduling is done by deletion of message from the buffer of the DTN nodes in the cluster, cluster

point/DropBox, carrier nodes and the LWC of the cluster point. These is done when transfer takes place between different types of nodes in the network. Detailed description is provided in Algorithm 7.

**Algorithm 7: Buffer Scheduling**

Step 1: **if** *(one DTN node encounters with another DTN node of the same cluster )* **then**
  Delete the message by checking their ACK list.
**end**
Step 2: **if** *( DTN node encounters with Cluster point/DropBox)* **then**
  Delete the message from DTN node after getting its Ack and updates ACk list for that message.
**end**
Step 3: **if** *(Cluster point/DropBox encounters with carrier node)* **then**
  Delete the message from the Cluster point/DropBox that has been sent to the carrier node and Delete the message also from the carrier node that has been sent to the Cluster point.
**end**
Step 4: **if** *(Cluster point/DropBox encounters with LWC)* **then**
  Delete the message from the Cluster point/DropBox that has been sent to the LWC.
**end**
Step 5: **if** *(LWC encounters with LWC(This LWC is towards internet gateway))* **then**
  Delete the message from the LWC that has been sent to the LWC towards the internet gateway.
**end**

### *F. Satellite Phone Module*

An important aspect of satellite phones is the transfer of acknowledgement messages between nodes. Hence, it became necessary to facilitate acknowledgement messages for satellite phones. Primary DTN routing strategies does not include transfer of acknowledgement messages. This needs the modification in the super class of the *routing* package, i.e. the *MessageRouter* class and also to all the other routing classes.
Another issue is the propagation delay in message transfer through satellite phones. As the geosynchronous satellites are at a distance of about 35,792 km from the Earth, there is always an approximate propagation delay in message transfer which is stated in algorithm 8 and algorithm 9. It can be calculated as follows;

$Distance\ from\ satellite = 35792km$
$Speed\ of\ light = 299762km/sec$
$Hence,\ Time = 35792/299762 = 0.12seconds(approx.)$
$Propagation\ Delay = Uplink\ Time + Downlink\ Time$
$= 0.12 + 0.12 = 0.24seconds$

**Algorithm 8: Satellite Module : Propagation Delay**

**if** *(satellite phone)* **then**
  transferDoneTime = SimClock.getTime() + ((1.0*messagesize) / transmitspeed) + (0.24)
**end**
**else**
  transferDoneTime = SimClock.getTime() + ((1.0*messagesize) / transmitspeed)
**end**

**Algorithm 9: Satellite Module : Emergency Message**

**if** *(other node in the connection is satellite phone and message is emergency message)* **then**
  Transmission Message to encountered satellite phone
**end**

This additional delay parameter is required to be included in the simulator for all nodes having satellite phones. This modification requires a change in the code of *CBRConnection* class. The *startTransfer* function in ONE was manipulated by setting the transfer time equal to the above value.
Cost of message transfer is quite high for satellite phones. So only some important messages had to be transferred through the satellite phones. This involved change in the code of *ActiveRouter* class. The restriction that regular messages won't be transferred through satellite phone requires proper implementation of our routing strategies. The *startTrasfer* function was modified to check if the message being relayed through a satellite phone is an emergency message or not. The function returns false for normal messages in case of satellite phone, otherwise follows the normal procedure. Creation of a new event in the simulator for important/emergency messages helps in ensuring this property.

Table I summarises both built-in and incorporated modules and their respective methods with basic functionalities for developing e-ONE simulator.

## III. Case Study:Hybrid Network Architecture for Post Disaster Management

In this case study, we describe a latency aware 4-tierd planned hybrid architecture [7][17]. Here, we have used DTN enabled smart phones, DropBoxes which may be high-end smart-phones or laptops, DataMules (e.g Ambulance, Boat etc.) which can move from one place to another and long-range WiFi Communication devices. In the next subsection we explain how we have used e-ONE to represent our communication architecture. Later, we analyze the results of simulations run using this architecture.

TABLE I.
REQUIRED MODIFICATION FOR E-ONE

| Modification | Files Modified | Remarks |
|---|---|---|
| Mobility Model | PostOfficeCluster Movement.java | Modification of Cluster Movement Model |
| Routing | ActiveRouter.java: starttransfer(), shouldsendMessage() | Restricted routing is performed for inter cluster |
| LWC Network formation and message transfer | Graph.java(File added to the package routing)<br><br>ActiveRouter.java: changedConnection(),startTransfer(), shouldsendMessage() | This is a class to perform basic functionalities of Graph & to form Tree Network.<br><br>This class contains methods for Graph and Tree network formation inheriting *Graph.java* Restricted routing is performed based on Tree information. |
| Fault Detection and Recovery | Graph.java<br><br>ActiveRouter.java: changedConnection() | Faulty node is detected by changedConnection() method. It obtains a new Graph and a tree network respectively. |
| Device Priority Scheduling | ActiveRouter.java | Higher priority nodes are served prior than lower priority nodes. |
| Buffer Scheduling | ActiveRouter.java starttransfer() Connnection.java finalizeTransfer() | The node which are recently involve in transfer of message are retrieve from finalize transfer of Connection.java. This is followed updatation and deletion of ACK List and message from the nodes. The function finalizeTransfer() is called by startTransfer() of ActiveRouter.java |
| Satellite Communication | MessageRouter.java & ActiveRouter.java | Transfer characteristics are implemented in MessageRouter.java, while the deciding a message transfer by ActiveRouter.java. |

### A. Post Disaster Communication Network Architecture

According to our perspective as shown in Figure 5., there is utilization of the low range and cheaper devices at the bottom layers and we have proceeded towards building the next higher ones with high range and costlier devices when the scenario cannot be handled by the lower layer. It has a fixed MCS to control centralized rescue/relief operations within affected area (AA) consisting of many shelter points (SPs). Rescue personnel within each SP carry smart phones which forms DTNs **[Tier-1]** to exchange information in the form of video clips, images, voice clips, and short text messages among

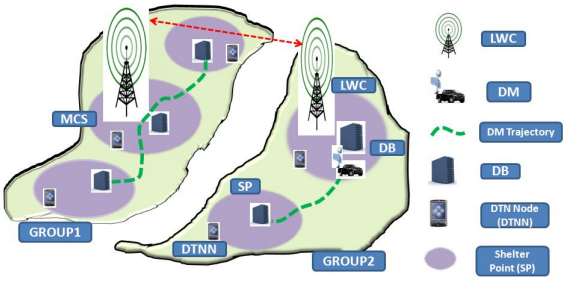



Figure 5. Four Tier hybrid Architecture using DTN Nodes,DBs,DMs and LWCs

themselves & deliver packets periodically to the nearest DB **[Tier-2]** belonging to each SP. As the DTNs formed are sparse, and DBs are far apart, we propose that, vehicles (i.e. boat, ambulance etc.) used by rescue/relief teams are equipped with Wi-Fi and VSAT (for emergency messages) and these act as DataMules (DMs) (mechanical back-hauls) **[Tier-3]** to carry information from DBs to MCS, within desired time L. If AA has a large diameter, deploying a dedicated DM per SP may not meet latency constraints and may also not be a feasible option. Hence we propose a grouping of DBs by using an efficient clustering algorithm. At the center of each such group, one (NLOS/near LOS)long range WiFi communicating device (LWC) **[Tier-4]**, accumulating data from a non-overlapping set of DBs, will be placed.

### B. System Model Overview

The SP corresponds to the vertex set, DB signifies the dropbox at each vertex or SP and the pathways among the SP correspond to the edge set of that graph G (V,E) where $V = \{SP_i\}; 1 \leq i \leq m$ and $E = \{E_{ij}|E_{ij}$ is pathway between vertices $SP_i$ & $SP_j$ ; $1 \leq i \leq m$ , $1 \leq j \leq m$ *and* $i \neq j\}$. Each vertex $DB_i$ has service time $S_T(DB_i)$. The graph G(V,E) is divided into $k$ sets of groups (*GR*). The vertex set of graph $G$ is partitioned into 2 sets: one is Group Centers ($GC$) and another is the set of Group Members ($GM$) where $GM$ =V$-GC$. Let $N$ be total number of data mules deployed and each $DM_{ip}$ has a distinct trajectory $T(DM_{ip})$.Let $k$ $LWC$s with range $R$ are deployed at each $GC_j$ subject to the following:

$V = GC \bigcup GM$

$DB_{ij}$ = $j^{th}$ DB in $i^{th}$ group $1 \leq i \leq k$ *and* $1 \leq j \leq m$

$DM_{ip}$ = $p^{th}$ DM in $i^{th}$ group $1 \leq i \leq k$ *and* $1 \leq p \leq n$

$GC_i$ = GC of $i^{th}$ group

$LWC_i$ = LWC of $i^{th}$ group

$T(DM_{ip}) \leq$ *Latency*

$Distance(LWC_i, LWC_j) \leq$ $R$ *if* $LWC_i$ and $LWC_j$ are connected.

The information packets have been differentiated into 2 types: (1) Packets generated at the DTN layer and destined to reach MCS (mostly), called *Relief Request Packet* ($R_e$); (2) Packets generated at $MCS$ intended for DTN-enabled devices, called *Relief Response Packets* ($R_s$). list of conversion used to model our architecture as shown

in Table II. All these packets need to be delivered within a predefined required latency. Here we present the worst case calculations pertaining to all 4 layers/Tiers for both types of packets, so that we can model the system in such a way that it guarantees 100% packet delivery.

TABLE II.
LIST OF CONVERSION USED TO MODELED THE ARCHITECTURE

| Variable Name | Meaning |
|---|---|
| $N$ | Total $DTN$ nodes surrounded by $DB$ |
| $g$ | Packet generation rate of $Re$ packes |
| $g'$ | Packet generation rate of $Rs$ packets |
| $p$ | Packet size of $Re$ and $Rs$ packets |
| $F_1$ | Maximum time of $DTN$ node to reach nearest $DB$ |
| $F_2$ | Time interval between 2 consecutive visits of $DM$ to the particular dropbox $DB$ |
| $F_3$ | Time interval between two consecutive visits of $DM$ to particular group center $GC$ |
| $L_d$ | Total load of $Re$ packets |
| $L_{d'}$ | Total load of $Rs$ packets |
| $DR$ | Data Rate |
| $S_U$ | setup Time |
| $S_T$ | Service Time |
| $L$ | Latency |
| $E_T(DB_{ij}, DB_{ik})$ | Travel Time by a DM from $DB_{ij}$ to $DB_{ik}$ without halting |

**Worst case latency calculation for Tier-1(T1) and Tier-2(T2):** Let $DR_{DTN-DB}$ be the data transfer rate from DTN to DB. $Q(R_e)$ be queuing delay for transfer of $R_e$ packets from any particular DTN to DB and $Q(R_s)$ be queuing delay of $R_s$ packet transfer from DB to DTN which is shown in figure 7,then:

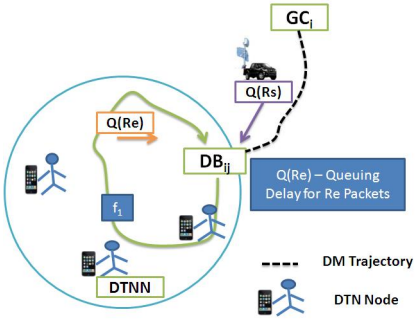



Figure 6. Tier-1 and Tier-2 Latency Calculation

$Q(R_e)$ = $F_1 \times g \times p \times (N-1)/DR_{DTN\_DB}$
$Q(R_s)$ = $L_{d'}(DB_{ij})/DR_{DTN\_DB}$ where $L_{d'}(DB_{ij})$ is total load of Rs packet at $DB_{ij}$ which is calculated in Tier-3.

In Worst case, the 1st generated $R_e$ packet come at the end of waiting queue at DB and it has to wait for other $R_e$ packets from $(N-1)$ DTN nodes to be offloaded at the DB if they come at same time and belongs ahead of 1st $R_e$ packet at queue . Also, packet bears delay because of setup time $S_U(DB_{ij})_{T1-T2}$ required at Dropbox between DTN and DB . Now, total service time required to serve a DTN at a particular $DB_{ij}$ between Tiers 1 & 2 will be:

$S_T(DB_{ij})_{T1-T2}$ =$Q(R_e) + Q(R_s) + S_U(DB_{ij})_{T1-T2}$

Now total service time $S_T(DB_{ij})_{T1-T2}$ is calculated between $T_1$ and $T_2$ using $Q(R_e)$, $Q(R_s)$ and $S_U(DB_{ij})_{T1-T2}$.
so $L_{T_2}$ = $F_1$ + $S_T(DB_{ij})_{T1-T2}$

**Worst case latency calculation for Tier-3 (T3):** At Tier-3, we consider the traversal of the packets using DMs. Here we assume that our DMs cover the DBs assigned to it in a circular manner as depicted in figure 7. Under such assumption, the DM serves each DB in its paths exactly once in one traversal. We now want to estimate the latency of a packet routed through $DB_{ij}$ to reach $GC_i$.

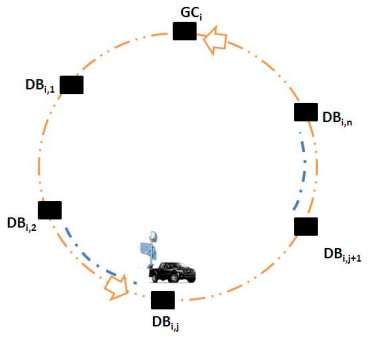



Figure 7. Circular DM Trajectory

At tier-3, the worst case may be a such scenario where a packet comes just immediately after a DM leaves the GC. In that case the packet has to wait almost the whole interval between two consecutive DM arrivals. Let $F_2(DB_{ij})$ be the time interval between 2 consecutive visits of $DM_{ip}$ to the particular dropbox $DB_{ij}$ and $F_2$ is also different for different $DB_{ij}$. $F_3(DM_{ip})$ is the time interval between two consecutive visits of $DM_{ip}$ to particular group center $GC_i$. Let $DR_{DB-DM}$ be the data transfer rate from DB to DM. DTN will dump packets at DB $(F_2/F_1)$ times. Then, Total data accumulated(Load) at dropbox $DB_{ij}$ will be:
$Ld(DB_{ij})$ =$F_2(DB_{ij}) \times g \times p \times N$.

Total data accumulated at dropbox $DB_{ij}$ for Rs packet will be $L_{d'}(DB_{ij})$ = $(F_3(DM_{ip}) \times g' \times p)/m$ where m is the total no of DB in Graph G(V,E). Also, packet at DB suffers delay because of set up time $S_U(DB_{ij})_{L2-L3}$ required at $DB_{ij}$ between DB and DM .
Total latency of Tier 3 should be included with the latency factor of previous layers and the total time of one or more DM visits to each of the DBs with waiting time between two consecutive visits which is basically the service time of other DBs coming into the DM trajectory. Hence, latency of Tier 3 comes as a function of Tier 2 latency, distance of DB from corresponding group center and service time of other DBs of that particular group. We have calculated total

latency $(L_3)$ using $F_2(DB_{ij})$ and $S_T(GC_i)*$. $F_2(DB_{ij})$ varies with the topological structure of the DM trajectory.

In such a case, DM travels in circular fashion, and thus, it serves exactly once in one traversal from $GC_i$ to current node and back to $GC_i$. Following is the mathematical model of circular trajectory:

$$L_{T_3} = L_{T_2} + F_2(DB_{ij}) + E_T(DB_{ij}, GC_i) + \sum_{j+1}^{1} S_T(DB_{ij})_{T2-T3} + S_T(GC_i)*$$

where

$$F_2(DB_{ij}) = E_T(DB_{ij}, GC_i) + E_T(GC_i, DB_{ij}) + \sum_{k=j}^{n} S_T(DB_{ik})_{T2-T3} + \sum_{k=1}^{j} S_T(DB_{ik})_{T2-T3}$$

**Worst case latency calculation for Tier-4 (T4):**Previous layers assures that $R_e$ packets have been delivered to GC of all groups. Now,it is the fourth layer that takes care of $R_e$ delivery to MCS. Similarly, Tier-4 also assures $R_s$ packets to be delivered to $GC$ of all groups.Thus, fourth layer modeling deals with latency incurred only due to $LWC$ interconnections. Since $LWC$s ,being layer 2 network devices ,don't allow loops in the network , we reduce the graph network , where LWCs behave as vertices and connection active between adjacent $LWC$s behave as Edges, into tree like network based on a suitable heuristic. For now, we consider minimum no of hops count from $MCS$ criterion to decide route to $LWC$ . After topology has been defined for Tier-4 , we need to model it in terms of Latency (12) fulfilling the below mentioned constraints.

**Constraints:**

(i)At any timestamp, if two LWCs (say $\text{LWC}_x$ and $\text{LWC}_y$) have their connections up, then neither $\text{LWC}_x$ nor $\text{LWC}_y$ can have connection up with any other LWC (say $\text{LWC}_z$) at the same time until and unless connection gets down.

(ii)However, any two LWCs (say $\text{LWC}_a$ and $\text{LWC}_b$) other than $\text{LWC}_x$ and $\text{LWC}_y$ can have connection up at the same time provided that $\text{LWC}_a$ and $\text{LWC}_b$ are completely non-overlapping to $\text{LWC}_x$ and $\text{LWC}_y$.

Given Tier 4 as tree like network with MCS as root and above mentioned constraints , our objective is to deliver Rs packets to GC and Re packets to $MCS$ with minimum latency.Since both types of RELIEF packets use the same network , they affect each other's latency . Thus, we use an optimal strategy that minimizes latency for both types of packets.

### C. Data Flow in PDCN using e-ONE Simulator

The fundamental to any planned approach is to design first and to test the performance of the design through some kind of dry run (simulation) before the actual deployment. In our case, we have a disaster-affected area for which two types of map are available to us. One map, which we call the *Path Matrix* for the simulation area, shows the SPs with possible vehicle connections between them. The other map, called the *Aerial Distance Matrix*, shows the SPs with Euclidean distance between them. The main design constraint in a post-disaster network design is the maximum allowed latency for the RELIEF packets. All the things mentioned above with few more relevant informations (shown in the flow diagram: Figure 8) are fed into a process called *Group Formation* which provide us with the complete deployment plan in which the SPs are divided into groups, one or more DMs are allotted to each group, the trajectories to each each DM allotted, the best placement of LWC towers inside each groups and the interconnection between the LWC towers. All the newly obtained group information along with original inputs are fed into our *e-ONE* simulator, which simulates the whole communication process and generate reports. The reports thus obtained are fed into our *Analysis Tool* to study the performance of our deployment plan. If the performance is found to be satisfactory, then it is ready for deployment.

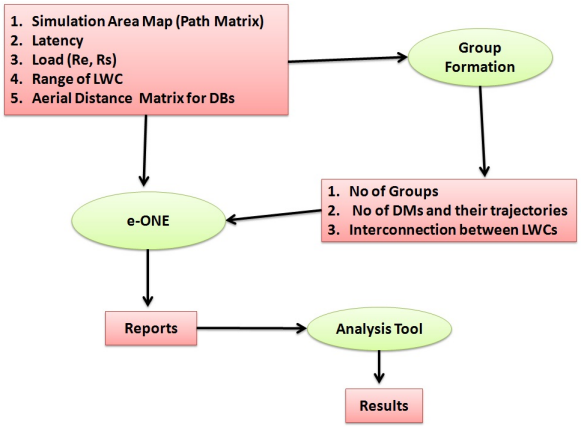



Figure 8. **Data Flow Diagram**

### D. Algorithm

What is apparent from the previous subsection is that the group formation is a key step in the design process and the main constraint to this design process is the maximum allowed latency to any RELIEF packet.Now the latency of a packet is sum of the latencies at different tiers. But majority of the latency is observed at Tier 3, where the packets are carried by DMs. So from maximum allowed latency we subtract the maximum possible latencies at other layers to get maximum allowed latency at Tier 3. With this modified latency $(L')$ value we form groups of SPs.

During group formation, we observe that a relief packet suffers maximum latency when it appeared to a DB immediately after the DM leave the DB, in which case the packet has to wait for a time equal to 3 times the the time required to travel from GC to corresponding DB including service time required at each intermediate DBs. Hence the maximum allowed time to reach a DB by a DM from from the GC is one third of $L'$.

**Algorithm 10: Main Function for Modeling Algorithm**

**Step 1: Input:** *MCS , L = Latency,* $\lambda$
// $\lambda$ is initial randomly chosen $T_4$ latency value
$L' = L - (F_1 + \lambda)$
// $L'$ is the maximum allowed latency for Tier-3
**Step 2: for** *(All unvisited DB)* **do**
  **Step 2.1:** GetGroups(MCS , GC , $L'$, G(V,E)) Returns the number of groups, Number of DMs for each group and their trajectories
  **Step 2.2:** Verify the feasibility of the group formation using our system model. If found not feasible, reduce $L'$ and go to step 2.
**end**
**Step 3:** Calculate Latency for Tier-4 using suitable heuristic and call it $\lambda'$ .
**step 4: if** *(*$\lambda' \leq \lambda$ *)* **then**
  Exit with Success.
**end**
**else**
  $\lambda = \lambda'$ //Replace initial $\lambda$ with calculated exact $\lambda$
  goto step 2
**end**

**Algorithm 11: Group Formation :: GetGroups(MCS , GC , $L$, G(V,E))**

**step 1:** GC = MCS
**step 2:** Deploy a DM and call it currentDM
**step 3: for** *(all DB's wich can be visited from GC in time* $\leq L/3$*)* **do**
  **step 3.1 : if** *(DB is already visited by some previous DM AND EdgeLength(GC,DB) for current DM < EdgeLength(GC,DB) for previous DM)* **then**
    visit DB by current DM
    Remove DB from the path of previous DM
    Update trajectory for previous DM
  **step 3.2: else**
    visit DB
    EdgeLength(GC, Next Unvisited DB) = EdgeLength(GC, DB) + min(EdgeLength(DB,Next Unvisited DB))
    DB = Next Unvisited DB
**step 4:if** *(There exist any unvisitd DB which can be traversed within L/3 from current GC)* **then**
  Go to step 2
**step 5:else**
  Deploy a new LWC at DB which can cover maximum number of unvisited DBs.
  That DB is included in GC set.
  Got to step 2.

We start by setting MCS as the first GC. We deploy a DM, and cover as many DBs as possible from that GC. If it covers all the DBs, we are through. If any more DB left, we deploy a second DM, if possible. Continue this way until no more addition of DM to the group can cover any more DB. If no more DBs left, we are through. Else, it is time to deploy LWC towers and add a new group. This starts by choosing best possible DB from the set of so far unvisited DBs as next GC and repeat the same process described above from the new GC. This process of forming group continues until no more DB left.

Our algorithm mostly follow a greedy approach. But the greedy approach has a tendency to stuck at a sub-optimal solution. To avoid this, we have taken a precaution as follows: if a DB can be reached by mor than one DMs, we assign that DM to the DB through which a GC can be reached in minimum possible time. Algorithm 10 and algorithm 11 describes the whole group formation procedure.

### *E. Customization e-ONE for PDCN*

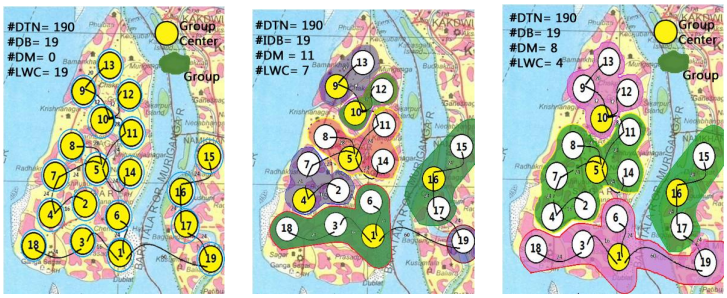
The PDCN architecture described above, being a hybrid architecture, requires several different features. The e-ONE can be utilized to provide these features to the architecture. Here, we describe how we have used the different features of e-ONE in PDCN.

*1) Mobility Model:* With the inclusion of the concept of the DropBox, it was evident that all the messages had to be relayed to it, as the DataMules could only establish connection with the DropBoxes. Hence, it became very important that all the nodes in the cluster should periodically visit the DropBox and drop the messages there. This was not the case for the Cluster Movement Model[ ] because the nodes in this movement were restricted to the cluster but followed Random Waypoint Model within. Hence, the periodic message dropping in the DropBox was not easy. Thus, the Post Office Cluster Movement Model was used to cope with this problem.

*2) New Modules:* Considering a disaster scenario, there might be some cases where emergency messages need to be delivered to the main control center as early as possible. The regular strategy is not particularly fast and thus satellite phones were used for relaying these emergency messages.

*3) Restricted Routing Strategy:* When the area was divided into various groups of clusters, there was a different type of restriction on the transfer of messages. Here the following restrictions were made to decrease the load in the system-

- If the message was for a node at the same cluster or at different cluster and the host was dtn node, (i) it would relay the message to the dtn node at the same cluster or (ii) if the dtn is in the range of DB then dtn not only will deliver message to the DB but also will receive ack from DB and this ack will be eventually propagated within the cluster for that particular message.
- If the message was for a node in the same cluster and the host was a DropBox, it would only relay the message to the DTN nodes.




(a) L=60 (b) L=180 (c) L=300

Figure 9. Number of Different Devices, Group Information for Latency 60,180 and 300 minutes of Planned Simulation

- If the message was for another cluster/shelter point in the same group and the host was a DropBox, it would only relay the message to the Data Mules.
- If the message was for another group and the host was a DropBox, it would relay the message to the LWC if the DropBox is in the group center, otherwise to the data mule.
- If the message was for another group and the host was a data mule, it would relay the message to the DropBox in the group center.
- If the message was for the same cluster/shelter point as the DropBox with which the host data mule is presently connected, it would relay the message to the DropBox.
- If the message is for the same group as the LWC, which is the host, then it would relay the message to the DropBox of the cluster it is in, otherwise to other LWCs.
- No data mule would relay a message to another data mule.

*4) Buffer Scheduling:* Buffering Scheduling is done by clearing of buffer from the nodes in the cluster, drop box, data mules and the LWC of the group center. These is done when transfer takes place between different types of nodes in the network.

- If one DTN node encounters with another DTN node of the same cluster then Delete the message by checking their ACK list.
- If DTN node encounters with dropbox then Delete the message from DTN node after getting its Ack and updates Ack list for that message.
- If dropbox encounters with data mules then (i) Delete the message from the dropbox that has been sent to the data mule and (ii) Delete the message also from the data mule that has been sent to the dropbox.
- If dropbox encounters with LWC then Delete the message from the Cluster point that has been sent to the LWC.
- If LWC encounters with LWC(This LWC is towards the MCS) then Delete the message from the LWC that has been sent to the LWC towards the MCS.

*5) Device scheduling:* Two types of devices encounters with dropboxes at each SP. One is DTN nodes & another one is DMs. Messages are uploaded and downloaded from both of them at DB. But a proper scheduling paradigm should be followed while both types of devices encounters DB at same time. DM should be served immediately with a higher priority than a DTN node. A miss arrival of DM will take longer time to get back again the opportunity of next arrival than DTN node to the particular DB. Because the roaming area of DM is larger than a DTN node. Henceforth DM is set to higher priority value than DTN Node.

### *F. Simulation Result and Analysis*

In this subsection first we have described simulation specification based on our architecture and then analyse the results.

*1) Simulation Setup:* Simulation is carried out using e-ONE Simulator [18] for the area of Sundarban, India; an area of 225sq.km is divided into 19 SPs as shown in Figure 9(a) with a density of 10 smart phones per SP, each having a data rate of 8Mbps and coverage range of 10m; nodes follow the Post Office Cluster movement model [16]. These nodes follow the *restricted epidemic routing* strategy for message transfer. They only interact with either other smart phones or the DB at the center. The velocity of DMs is restricted to 10 km/hr. These DMs move (trajectory) between the group center SP to the other SPs which is also get from the modeling . The pause time at each SP for the DM is set from the modeling. The DMs are restricted to interact only with the group DBs and take messages only if it is entitled for some device outside the cluster it is presently in. The LWCs at the group centres have a coverage range of 9 kms. The data-rates for each type of device had been set based on lab-based experimental values. list of simulation parameters for our modeling as shown in Table III.

*2) Simulation Settings:* The use of e-ONE required some new settings to be included in the settings file for the ONE. In this section, we present the new settings used in e-ONE for PDCN.

**Starting node number of all types of nodes**

Group.first(type of Device)= first address of the node.

The networks needs to know the first address of the type of device implemented.

e.g.:- Group.firstBT = 53(first address of the DTN node)

Group.firstDM = 38(first address of the Data Mule)

**Setting of total no of data mules**

Group.DMS=n

TABLE III.
SIMULATION PARAMETERS USED TO SIMULATE THE ARCHITECTURE

| Parameter | value |
|---|---|
| Total Area | 225 sq.km |
| No of Shelter Points | 19 |
| Area surrounded by SP | 12 sq. Km |
| No of DTNs/SP | 10 |
| Latecncy | 200 minutes & 240 minutes |
| LWC Range | 9 KM |
| Data Rate of DTN node | 2 Mbps |
| Data Rate of DB to DM | 20 Mbps |
| Data Rate of DB to LWC | 18 Mbps |
| Data Rate of LWC to LWC | 8 Mbps |
| Mean $F_1$ | 20 Minutes |
| Mean $R_e$ Packet Generation Rate of DTN Node | 10 Packets/DTN/Hr |
| Mean $R_s$ Packet Generation Rate of MCS | 3 Packets/minutes |
| Simulation Time | 20 hours |
| Mean Packet Size | 1 MB |

Total no. of Data Mules in the network is n. E.g.:- Total no. of data mules for 200 latency is 9. Therefore in the Simulation setting it is give as below
Group.DMS=9
**DataMule Trajectory**
Group.DM(i)=a1, a2, a3
The above line says that Data Mule (DM) visits a1, a2, a3, Drop Box. It means that the Data Mule starts from Cluster Center and visits a1 Drop Box then a2 Drop Box and so on. i is the Data Mule no. starting from 1 to n. E.g:-
Group.DM1= 11
Group.DM2= 7,8
DM1 starts from cluster center and visits 11 no. Drop Box. DM2 starts from cluster center and visits 7 and 8 Drop Box.
**Node number of the group center dropboxes**
*Dropboxes range from 19 to 37. Such that, 1 contains DB19 and similarly 19 contains DB37*
*Centres for 200AT are 1,4,5,10,13,16*
$Group.group\_centers = 19, 22, 23, 28, 31, 34$

*3) Planned Deployment:* Figure 9 illustrates the effect of latency (L) on the process of group formation. We have also observed as we increase the value of L lesser the number of groups have been formed compared to lower latency. The size of a group is directly proportional to the coverage area of the DMs within L. The more is the value of L the more area will be covered by DMs resulting less number of group formation as shown in figure 9 (a) (b) and (c). The planned approach actually yields 100% packet delivery as shown in figure 10. The figure also shows that nearly 30% of all the packet from the system are delivered nearly within 25 minutes. We feel that these are from the DTN nodes near the GCs as well as the MCS. Again we observe that nearly 70% packets are delivered within the time equal to half the L.

Figure 11 illustrates mean delay at every tire with corresponding error bar highlighted in them. The observation suggests that majority of the delay is contributed by the traversal of the packets from IDBs to the respective GCs through the DMs.

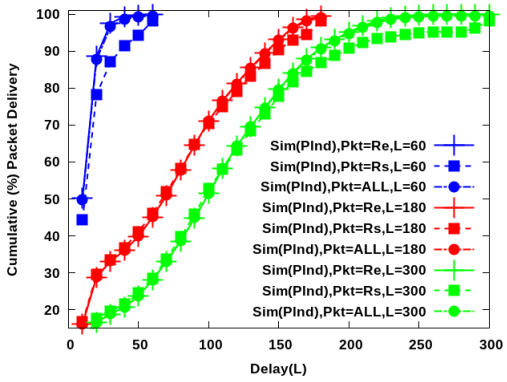



Figure 10. Cumulative Packet(%) Delivery for L=60, 180 and 300 of Planned Simulation

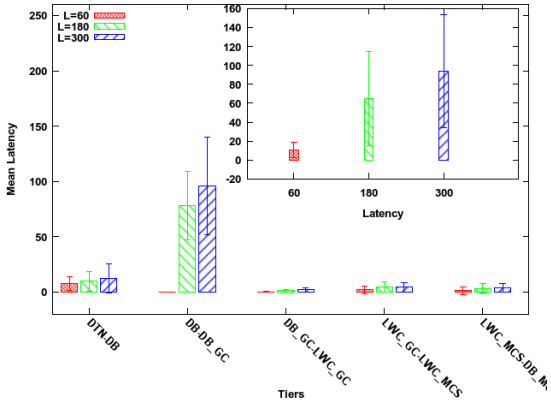
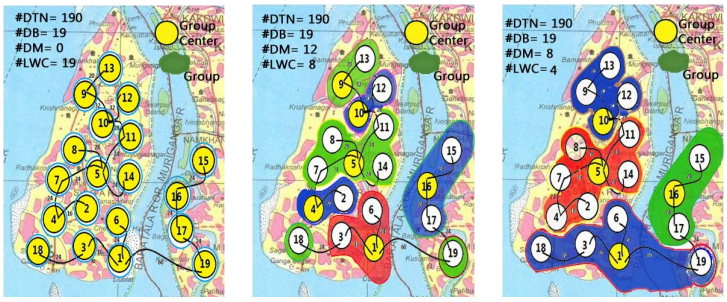



Figure 11. Mean Latency with error bar for L=60,180 & 300 of Planned Simulation

*4) Unplanned Deployment:* Parameters and logic of group formation are almost same for both planned and unplanned approach but for planned deployment the value of parameters are determined through some system model where as those are randomly taken some presumed values which can not be (service time) determined without mathematical model. We have also obtained same co-relation between no of groups and L which is shown in Figure 12 after some certain value of $L_0$ but the nature of relation between no of groups and L may be random for any value lesser than $L_0$.

Around 10% to 20% packet loss is observed in every case under unplanned solution, as seen from figure 13. We also noticed mean latency with error bar is always high compared to the planned simulation as shown in figure 14.

Figure 15 shows the lay out of our packet analysis tool which analyze performance (e.g cumulative packet delivery of different types of packet, Avarage latency , Tier wise latency). If packets are lost then our tool is capable of identifying the Tier/layer is responsible for that loss. It is also used to show the trace and associated graph of dropped packets. On selection of a dropped packet from




(a) L=60 (b) L=180 (c) L=300

Figure 12. Number of Different Devices, Group Information for Latency 60,180 and 300 minutes of Unplanned Simulation

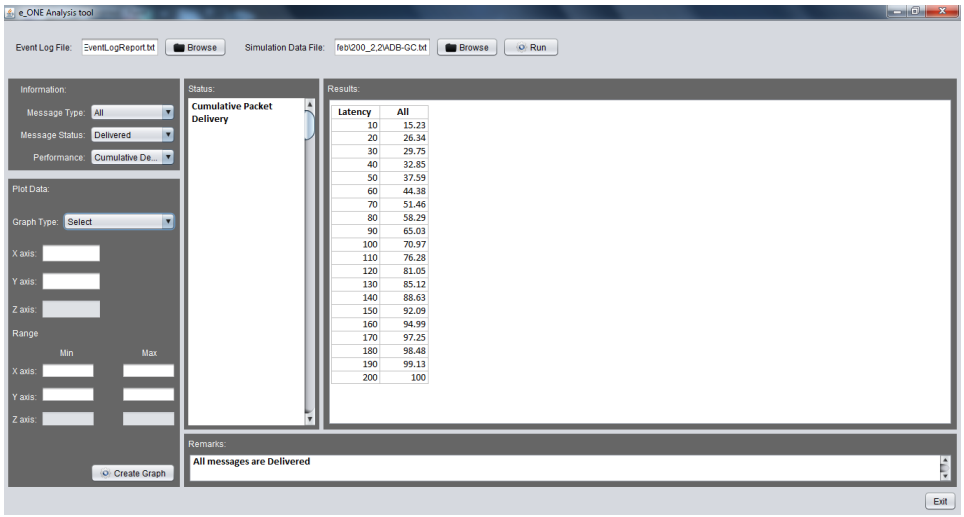



Figure 15. e-ONE Packet Analysis Tool

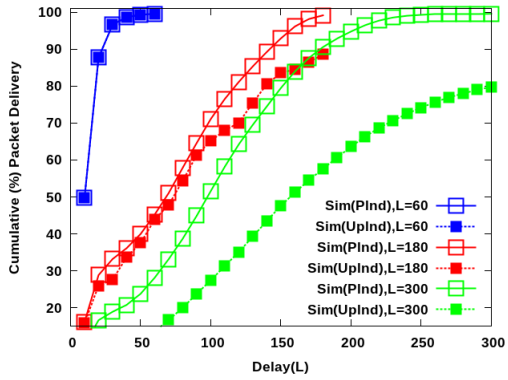



Figure 13. Mean Latency with error bar for L=60,180 & 300

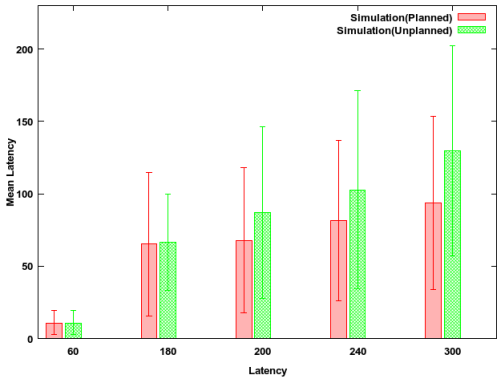



Figure 14. Cumulative Packet Delivery for L=60,180 & 300

its list, the tool shows the message path and possible reasons for packet dropping along with corresponding trace history. It also helps to draw different types plots for analysis the results.

## IV. Conclusion & Future Work

Our proposed e-ONE simulator offers a heterogeneous or challenged network evaluation system with a variety of enhanced modules like mobility pattern, infrastructure network system (MESH, SAT, LWC), intelligent routing strategy, device scheduling, buffer scheduling etc. In this paper we have shown cent percent packet delivery within the given latency by our latency aware post disaster management architecture ; evaluated through e-one. We have also noticed that 70% to 74% packets are being delivered with half of the latency factor.

Future extension of e-ONE is to overcome a few limitations which still exist, such as,

(1) Shape of the cluster is a polygon in case of a real life scenario but here we have considered it as a circle

(2) Define movement pattern which is more realistic for post disaster management inside the cluster

(3) Combination of our Network Resource Allocation Software [19] and e-ONE for better post disaster network analysis & management;
(4) Deployment of an architecture in one of the disaster prone areas of the Sundarbans.

## ACKNOWLEDGEMENT

The authors are grateful to the anonymous reviewers for constructive suggestions and insightful comments which greatly helped to improve the quality of the manuscript. This Publication is an outcome of the R&D work undertaken in the ITRA project of Media Lab Asia entitled Post-Disaster Situation Analysis and Resource Management Using Delay-Tolerant Peer-to-Peer Wireless Networks (DISARM). (Ref. No.: ITRA/15 (58) /Mobile/DISARM/01).

## REFERENCES

[1] K. R. Fall, 'A Delay-Tolerant Network Architecture for Challenged Internets', *Proc.* ACM SIGCOMM (2003).
[2] A. Pentland, R. Fletcher and Hasson , 'DakNet: Rethinking Connectivity in Developing Nations', IEEE Computer, vol. 37, no.1, (2004).
[3] T. Hossmann, F. Legendre, P. Carta, P. Gunningberg, and C. Rohner,'Twitter in disaster mode: Opportunistic communication and distribution of sensor data in emergencies',ExtremeCom (2011).
[4] B. Braunstein, T. Trimble, R. Mishra, B. Manoj, L. Lenert and R. Rao, 'Challenges in using of distributed wireless mesh network in emergency response',3rd International ISCRAM Conference (2006).
[5] A. Seth, D. Kroeker, M. Zaharia, S. Guo and S. Keshav, 'Low-cost Communication for Rural Internet Kiosks using Mechanical Backhauls',Proc. MobiCom 2006, Los Angeles, USA (2006).
[6] S.M. George, W. Zhou, H. Chenji et al. 'DistressNet: A Wireless Ad Hoc and Sensor Network Architecture for Situation Management in Disaster Response',IEEE Communications Magazine, Volume: 48 , Issue: 3 (2010).
[7] S. Saha, V.K Shah, R. Verma, R. Mandal and S. Nandi, 'Is It Worth Taking a Planned Approach to Design Adhoc Infrastructure for Post Disaster Communication?'in Proceedings of the ACM CHANTS12, co-located with MobiCom 2012, Istanbul, Turkey (2012).
[8] T. Hossmann, F. Legendre, P. Gunningberg, C. Rohner,'Twitter in Disaster Mode: Opportunistic Communication and Distribution of Sensor Data in Emergencies'ExtremeCom2011, September 26-30, 2011, Manaus, Brazil.
[9] ONE: 'www.netlab.tkk.fi/tutkimus/dtn/theone/'.
[10] DTNSim2 :'http://www.codeforge.com/article/142062'
[11] A. Vahdat and D. Becker ,'Epidemic routing for partially connected ad hoc networks', Technical Report CS-200006,Duke University (2000).
[12] A. Lindgren, A. Doria and O. Schelen , 'Probabilistic Routing in Intermittently Connected Networks',SIGMOBILE Mobile Computing Communications Review (2003).
[13] T. Spyropoulos et al., 'Spray and wait: an efficient routing scheme for intermittently connected mobile networks',ACM SIGCOMM workshop on Delay-tolerant networking (2005).
[14] T. Spyropoulos et al., 'Spray and Focus: Efficient Mobility-Assisted Routing For Heterogeneous & Correlated Mobility',in Proc. Fifth IEEE PERCOM Workshops, 2007.
[15] J. Burgess et al., 'Maxprop: Routing for vehicle-based disruption-tolerant networking',in Proc. INFOCOM (2006).
[16] M. Romoozi et al.,'A Cluster-Based Mobility Model for Intelligent Nodes',at Proceeding ICCSA '09 Proceedings of the International Conference on Computational Science and Its Applications: Part I (2009).
[17] S. Saha, S. Nandia, P. S. Paul, V. K. Shah, A. Roy, S. K. Das, 'Designing delay constrained hybrid ad hoc network infrastructure for post-disaster communication', Journal of Ad Hoc Networks, Elsevier (DOI: 10.1016/j.adhoc.2014.08.009), 2014.
[18] http://www.nitdgp.ac.in/MCN-RG/eONE/eONE.html.
[19] S. Saha, N. Agarwal, P. Dhanuka and S. Nandi,'Google Map Based User Interface for Network Resource Planning in Post Disaster Management',in Proceedings of the ACM DEV 2013,co-located with COMSNETS 2013, Bangalore, India.

**Sujoy Saha: He is presently a faculty member in the Department of Computer Applications, National Institute of Technology, Durgapur, India. He has completed B. Tech (Computer Science and Engineering) from NIT Calicut and M.Tech (Computer Science And Engineering), Jadavpur University in 2005. He is presently pursuing his Ph.D in designing Secure Resource Constrained DTN Architecture for Challenged Scenario under Dr. Subrata Nandi, Department of CSE, NIT Durgapur. His areas of research are Mobile Ad-hoc Network, Network Modeling, Delay Tolerant Networks and Network Security. He has published 9 papers in different International Conferences/Journals which includes leading conferences ACM Mobicom, ACM DEV, ICDCN, Journal of Ad Hoc Networks in Elsevier etc. He is also Project Co-Investigator of the project DISARM, Funded by Media Lab Asia /ITRA, MCIT, Govt. of India.**

**Rohit Verma: He is presently working at Schneider Electric India. He completed his B.Tech. in Computer Science and Engineering from National Institute of Technology, Durgapur, India in the year of 2013. He worked as a young researcher in a project entitled Post-Disaster Situation Analysis and Resource Management Using Delay-Tolerant Peer-to-Peer Wireless Networks (DISARM) which is funded by ITRA, Media Lab Asia, Government of India in 2013. He worked on security of Delay Tolerant Network & enhanced ONE simulator. He has published five international conference papers like in MOBICOM CHANTS workshop.**

**Partha Sarathi Paul: He is presently working as a Senior Research Fellow in the Department of Computer science & Engineering at National Institute of Technology Durgapur, India. He has received his B.Sc. degree from University of Calcutta with honours in Mathematics in the year of 1998. He then got his M.Sc. degree from the same university in pure mathematics in the year of 2000. After that he completed his M.Tech. in Computer Science program form Indian Statistical Institute, Calcutta in the year of 2003. Presently he is pursuing Ph.D. in the area of Modeling, Design & Analysis of Hybrid Networks Infrastructures, under Dr. Subrata Nandi at NIT Duragpur. His research interests include Mobile Ad-hoc Network, Network Modeling, Delay Tolerant Networks, etc.**

**Somir Saikia: He is presently working at Vantage Circle. He has got his M.C.A (Master of computer applications) from National Institute of Technology, Durgapur, India in the year of 2014. He worked as a young researcher in a project entitled Post-Disaster Situation Analysis and Resource Management Using Delay-Tolerant Peer-to-Peer Wireless Networks (DISARM) which is funded by ITRA, Media Lab Asia, Government of India.**

**Subrata Nandi: He is presently working as Associate Professor in the Department of Computer Science & Engineering, NIT, Durgapur,India. He has completed B.Tech (Computer Science & Engineering) from University of Calcutta and M.Tech (Computer Science & Engineering) from Jadavpur University. He has completed PhD (Titled: Information Management in Large Scale Networks) from the Department of Computer Science And Engineering, Indian Institute of Technology, Kharagpur in 2011. He worked as a project fellow in the Indo-German (DST-BMBF) project during 20082010. He visited Dept. of High Performance Computing, TU Dresden, Germany as visiting research scientist and received ACM SIGCOMM travel grant in 2008. He has published 25 papers in different International Conferences/Journals which includes Physical Review E, ACM Mobicom, ACM SIGCOMM, ACM DEV, etc. His broad interest lies in developing technologies for developing regions with specific focus on Peer to Peer Network, Mobile Ad-hoc Network, Delay Tolerant Network, Service-oriented Architecture, etc. He is also Project Investigator of the project DISARM, Funded by Media Lab Asia /ITRA, MCIT, Govt. of India.**