DTN DESIGN AND IMPLEMENTATION

1. Introduction

This document shall outline the prototype design and implementation of DTN (Delay Tolerant Networking) message exchange and decision choice for routing / forwarding based on DTN metric collected.

Prototype Design Goals

This prototype of DTN is built with a design goal of providing a GSTAR capable router to make forwarding decisions when it receives a packet destined to a node that does not belong to intra-partition graph.

2. Implementation Details

This section shall outline the current implementation (version 1.0) details. This may need to be updated with the further design changes.

2.1. Initialization

DTN Table needs to be built across long periods of time. When two nodes come in contact how we can possibly initiate a DTN LSA Message exchange process?

- A node can start by adding itself onto DTN Table to be the first entry and sending its own availability (probability) to other reachable nodes
    - Example – Node A with **guid** "120" shall begin with an entry to its **DTN Table** as

GUID	**TimeStamp**	**HashTable<GUID, connection probability>**
120	Current time stamp	120,0.9 (only one entry)

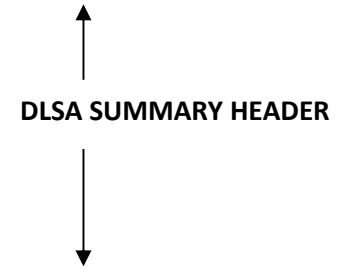
- When neighbor table changes – First time the entry is added to neighbor table
- When a node receives a DLSA message from one of its neighbors and updates DTN Table, needs to be propagated to "other" neighbors.

2.2. Message Exchange

2.2.1. Message Format

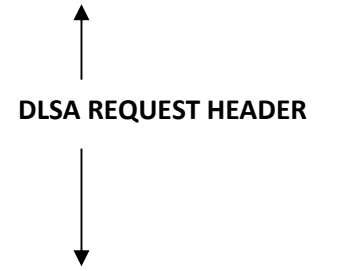
1. **DLSA SUMMARY**

Type 0x0001 (uint32)
Source GUID (uint32)
Destination GUID (uint32)
DLSA Summary Vector< GUID:Timestamp>
Vector < GUID : TimeStamp>



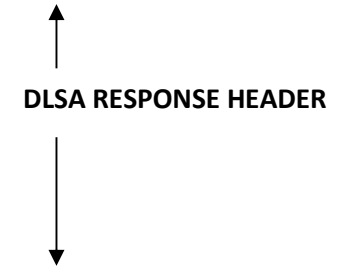
2. **DLSA REQUEST**

Type 0x0002 (uint32)
Source GUID (uint32)
Destination GUID (uint32)
DLSA Summary Vector< GUID>
Vector <GUID >



3. **DLSA RESPONSE**

Type 0x0003 (uint32)
Source GUID (uint32)
Destination GUID (uint32)
Current Time stamp
HashTable< GUID : **TimeStamp : HashTable<GUID:metric>** >



2.3. Storage Data structure

DTN Table shall be maintained as a **nested Hash Table** as below

Hash Table< K1 , V1>

K 1: GUID

V1: [Time Stamp, **Hash Table<K2,V2>**]

Time stamp represents the time at which DLSA summary was received from node (K – GUID)

Hash Table<k2,v2> : represents a hash table with GUID and metric field.

Where, metric currently used is connection probability.

Performance: O (1) Look up

2.4. Routing/Forwarding

Routing decisions are made on two fold information -

Intra partition graph information (LETT/SETT/Storage capability)

Inter partition graph information (connection probability)

GUID	Time stamp	GUID:Metric
120	Timestamp at which dlsa summary was received	120 :0.9 130 :0.5 140 : 0.6
121	"	121:1.0 123:0.6
125	"	125:0.8 131:0.7 141:0.6

- If a destination is not found on the intra-partition graph, router shall look-up the DTN table.
- We aim to discover node or candidate nodes which have possibly come in contact with the destination at some point.
- Next, we could forward packet to this intermediate node that's a part of intra-partition graph that can possibly come in contact with destination and forward the packet.
- We could get a set of the intra-partition graph node GUID's from the neighbor table and check them against the DTN table.
- For every node in the intra partition graph we could check for the destination node in the DLSA summary information (GUID : Metric) part of DTN table , to extract a set of intra-partition nodes that are candidate intermediate nodes this packet is to be forwarded to.
- Metric is to be used as deciding factor for the selection of such intermediate nodes.
- When destination does not belong to intra-partition graph –

Source → R1 (DTN Table look-up) → Intermediate node (belongs to intra-partition graph) → Destination

3. CLICK Implementation

Elements implemented

- **mf_dtntable**

  Contains all the API's for operations to be performed on DTN Table. The operations include building a summary hashmap , comparing summary hashmap with DTN table entries, updating new DLSA entries onto DTN table, making a subset of DTN Table based on request vector

- **mf_senddlsasummary**
  - Sends dlsa summary packet (hashmap of guid and timestamp)
- **mf_processdlsasummary**
  - process dlsa summary packet received
  - send dlsa request if needed (vector of guid)
- **mf_processdlsarequest**
  - process dlsa request packet
  - send dlsa response packet (subset of dtn table information)

- **mf_routedecision**
    - Uses the routing table and DTN table to make decisions on forwarding.
    - Essentially returns next hop for a given destination

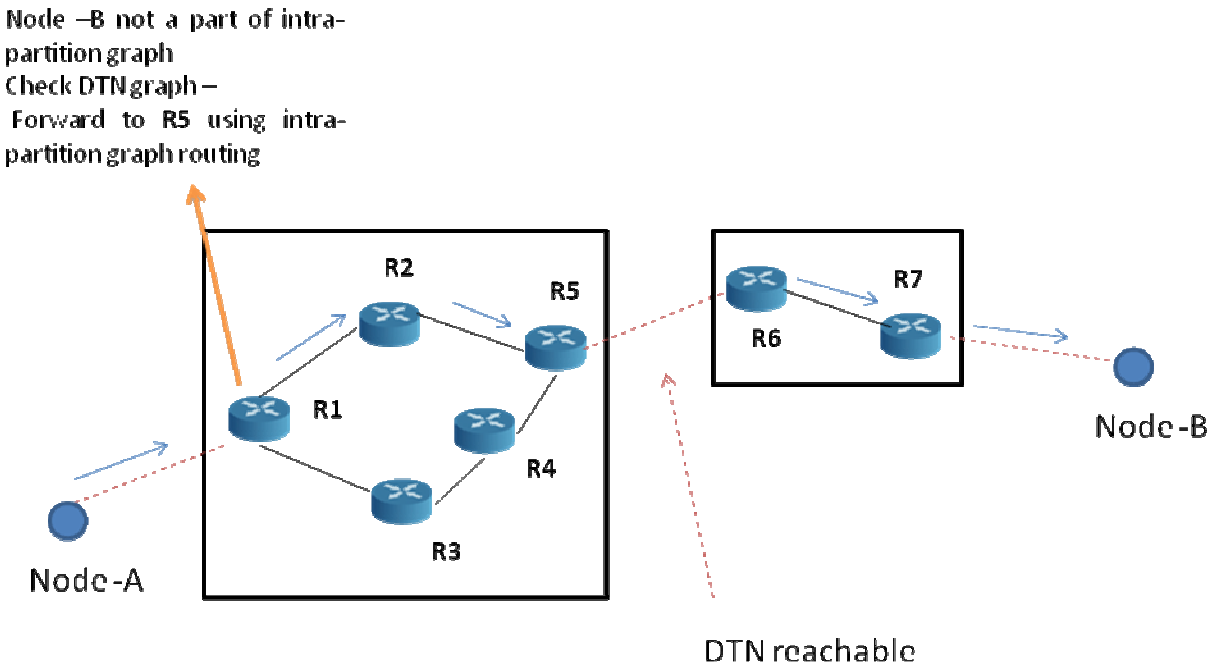
** Check the inline documentation in the CLICK scripts for further description*

4. Integration and Possible Application Demo

ORBIT DEMO

SCENARIO -1

A topology may be pushed onto orbit nodes. This may be two intra partition graphs and connectivity that may be down at some point



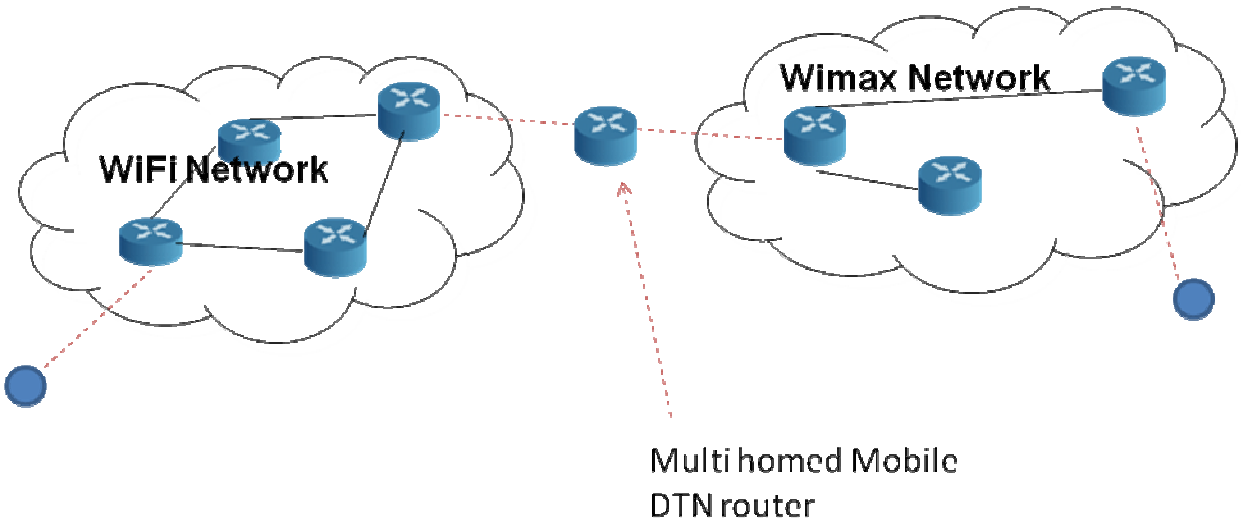
Split a given topology into two intra partition graphs and forward the message from one to another as shown above.

SCENARIO -2

Between two domains

WiFi - Wimax network – A multi home DTN device acting as carrier between these two islands of networks

May be used to demonstrate the use of DTN capability in GSTAR



5. Further

Some of the questions to be answered …

Duration over which the connectivity probability is to be measured

- *may be based on scenarios to be addressed?*

How do we identify a DTN capable router could ferry data to internet (not another adhoc network)

(Rural network scenario)

- *could ping and determine its connectivity through a certain network to internet?*
- *feedback from DTN node about its connectivity?*

Capacity of a DTN router (Storage)

- *Depending on scenario?*

How do we determine what is to be discarded from the DTN table (timely clean up)

*If destination is **not found** in intra partition graph and DTN table ?*