Architecture of DtnSim

Benoit Coeugnet
INSA Lyon - 4TC
benoit.coeugnet@insa-lyon.fr

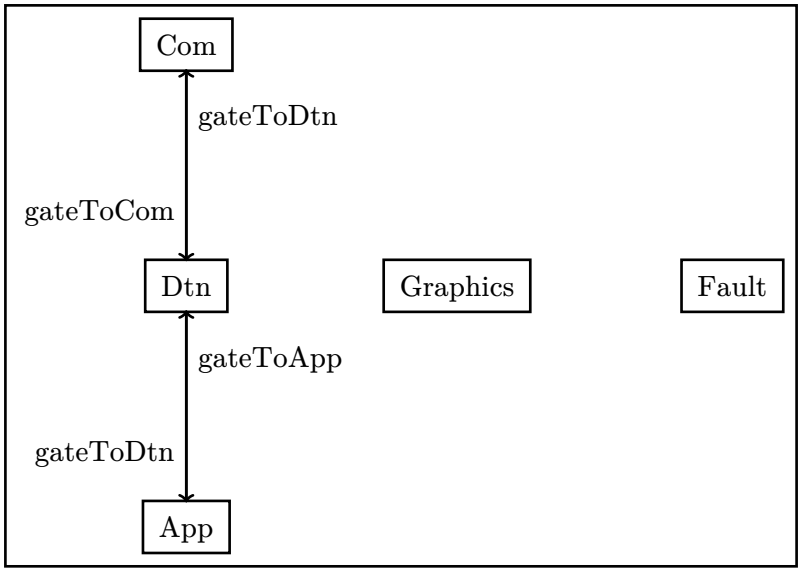
Table des matières

At the end of the document, each bold word in the document is explained.

I Graph view of the simulator

DtnSim

Node



II Parameters

II.1 DtnSim

The general architecture of the network.

Submodules :

- Central node
- Nodes

II.2 Nodes module

Submodules :

- App
- Dtn
- Com
- Graphics
- Fault

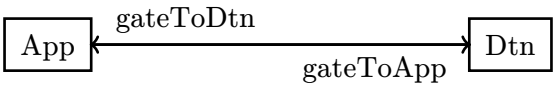
II.3 Central module

Parameters :

Type	Parameter	Default value	Description
bool	saveTopology	**false**	Store the topology in a .dot file.
	saveFlows	**false**	Store the flow in a .dot file.
	saveLpFlows	**false**	Store the **linear programming** flow in a .dot file.
	useSpecificFailureProbabilities	**false**	Whether or not to use specific failure probabilities. This determines whether or not a set of contacts can fail over time.
	useCentrality	**false**	2 way of removing contacts if needeed : randomly or by chosen the ones with the maximum **centrality.**
	faultsAware	**true**	If true, faulty contacts are removed from the contact plan (so better routing decisions can be made)
	useUncertainty	**false**	To use Opportunistic CGR or not.
	enableAvailableRoutesCalculation	**false**	Calculate and emit statistics about routes in a network.
string	contactsFile	"contacts.txt"	Where the algorithm can find the contact plan.
	contactIdsToDelete	""	A list of contact to delete.
	collectorPath	""	Define the path for the **metricCollector** to store it data.
int	deleteNContacts	0	A number of contact to delete.
	mode	1	Define in which opportunistic contacts are included in the simulation, 0 for no opportunistic contacts, 1 for regular contact discovery, 2 for complete knowledge in advance.
	repetition	0	Number of run in the simulations
double	failureProbability	0	Probability of failure of any contact in the contact plan.

II.4 App sub-module

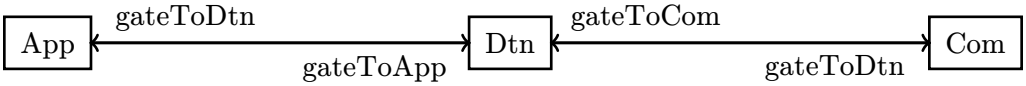
Links :



Type	Parameter	Default value	Description
bool	enable	**false**	Enable the generation of traffic (better if it's enable).
	returnToSender	**true**	Return the bundle to the sender if there is a problem.
	critical	**false**	Set if a bundle is critical or not.
	custodyTransfer	**false**	If **custody** transfer is enabled or not for the generated bundles.
string	destinationEid	"1"	Give a destination EID for the bundles generated.
	bundlesNumber	"1"	Define the number of bundles to generate.
	size	"1024"	Define the size of a bundle in bytes.
	start	"0"	Define the time at which the message generation starts.
	externalTrafficEvents	""	To generate external traffic (that are not between nodes in the topology).
double	interval	0	The interval between the generation of each bundles.
	ttl	9000000	Time to live of a bundle.

II.5 Dtn sub-module

Links :

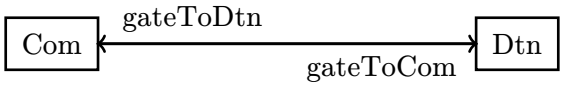


Type	Parameter	Default value	Description
bool	printRoutingDebug	**false**	For debug purpose
	saveBundleMap	**false**	Create a .csv file with statistics about bundles (like source, destination etc)
string	routing	"direct"	Switch between different routing solutions.
	routingType	"none"	I assume it's the same as frouting but only for RoutingCgrModelRev17 routing.
	frouting	""	Path to file which encodes routing decisions (only valid for BRUF routing)
	ts_start_times	""	Specify the timestamp for sending the bundles. *Not compatible with ts_duration.*

int	bundlesCopies	1	The number of copies of each bundles.
	sdrSize	0	SDR Memory Size in Bytes (0 = infinite)
	ts_duration	−1	Waiting time between sending two bundles. *Not compatible with ts_start_times.*
double	sContactProb	1.0	To set a probability of contact if a link is in the contact plan (only in RoutingCgrModel350_Probabilistic).
	pEncouterMax	0.7	Used to know to which node to forward the message to (only in RoutingProPHET)
	pEncouterFirst	0.5	Initial value to link for the first time (only in RoutingProPHET)
	pFirstThreshold	0.1	If the connexion has already happened (only in RoutingProPHET)
	ForwThresh	0	Forward Threshold : a bundle is forwarded if the predictability is greater than the threshold (only in RoutingProPHET)
	alpha	0.5	Parameter for RoutingProPHET to calculate the delivery predictability.
	beta	0.9	
	gamma	0.999	
	delta	0.01	

II.6 Com sub-module

Links :



Type	Parameter	Default value	Description
double	packetLoss	0.0	Set the probability of a packet being lost. Checked when the packet arrives.

II.7 Graphics sub-module

Type	Parameter	Default value	Description
bool	enable	**true**	If true, enable graphical elements to be displayed as red nodes when there is an error or as a line between nodes that are in contact.

II.8 Fault sub-module

Type	Parameter	Default value	Description

bool	enable	**false**	Enable fault mode.
int	faultSeed	0	?
double	meanTTF	0s	Define mean time between failures
	meanTTR	0s	Define mean time between recovery

III Outputs

III.1 Recorded explanation

The recorded column contains information about the type of the record. For example, dtnBundleSentToCom only records a count value, so at the end of the simulation, only the total number of bundles sent will be available. But with the dtnBundleSentToAppHopCount, the data will be available as a vector or histogram.

There are 2 ways to save the output, in two different file formats:
- Vector (.vec) - has data for every second of the simulation (so data depending on time)
- Scalar (.sca) - has data for the whole simulation such as *count, sum, max, min, mean.*

At the end of the simulation, the scalar file contains only one number for each of the output parameters and data type:
- count : count the number of times a signal is received
- sum : sum of the values received
- max : store the maximum of the values received
- min : store the minimum of the values received
- mean : process the mean of the values received

III.2 Central module

Output	Recorded	Description
contactsNumber	sum	The total number of contacts in the contact plan.
totalRoutes	sum	It's the total number of routes from all nodes to all nodes.
availableRoutes	sum	Same as totalRoutes, but without the contacts that have been deleted using the *deleteNContacts* or *contactIdsToDelete* parameter and those that have failed (*failureProbability* parameter).
pairsOfNodesWithAtLeastOneRoute	sum	The number of pairs of nodes that have at least one route between them.

III.3 Dtn sub-module

Output	Recorded	Description
dtnBundleSentToCom	count	Count the total number of bundles sent from the Dtn sub-module to the Com sub-module.
dtnBundleSentToApp	count	Count the total number of bundles sent from the Dtn sub-module to the App sub-module.
dtnBundleSentToAppHopCount	vector, histogram	Record the hop count of the bundle sent to the App sub-module.

dtnBundleSentToAppRevisitedHops	histogram	Subtract the number of hops from the number of visited nodes to get the number of revisited hops.
dtnBundleReceivedFromCom	count	Count the total number of bundles received by the Dtn sub-module from the Com sub-module.
dtnBundleReceivedFromApp	count	Count the total number of bundles received by the Dtn sub-module from the App sub-module.
dtnBundleReRouted	count	Count the total number of bundles that had to be re-routed due to the end of a transmission.
sdrBundleStored	vector, timeavg, max	All bundles in the simulation, stored in all SDR queues.
sdrBytesStored	vector, timeavg, max	All data in Bytes stored in all SDR queues.
routeCgrDijkstraCalls	vector, sum	To track the number of times Dijkstra is called to find the best route.
routeCgrDijkstraLoops	vector, sum	To track the number of iterations performed by the algorithm.
routeCgrRouteTableEntriesCreated	vector, max, sum	Track the number of routes that are added to the routing table (find by Dijkstra)
routeCgrRouteTableEntriesExplored	vector, max, sum	Number of explored routes (including filtered routes, i.e. pessimistic) taken at destination nodes.

III.4 App sub-module

Output	Recorded	Description
appBundleSent	count	Count the total number of bundles sent from the App sub-module to the Dtn sub-module.
appBundleReceived	count	Count the total number of bundles received by the App sub-module from the Dtn sub-module.
appBundleReceivedHops	mean, max, min, histogram	Track the number of hops that a bundle has made before reaching the App sub-module.
appBundleReceivedDelay	mean, max, min, histogram	Track the time between the creation of the bundle and its receipt by the app sub-module.

IV Definition

Centrality : In this case, the algorithm calculates for each node the shortest path to each node. The centrality of a node is then calculated by dividing the number of shortest paths through it by the total number of shortest paths.

Custody : In DTN networks, nodes can take custody of a bundle. This means that the node acts as if it's the source of the bundle. For example, it can retransmit the packet if it's lost, without having to start all over again.

Linear programming : In a problem with linear relations, it's a way of getting the best result, such as maximum flow in our context. It's also known as linear optimisation.

Metric Collector : It basically gather data about nodes, routes, bundles…