Congestion Control in Disruption-Tolerant Networks: A Comparative Study for Interplanetary and Terrestrial Networking Applications

Aloizio P. Silva[a,*], Scott Burleigh[b], Celso M. Hirata[a], Katia Obraczka[c]

[a]*Instituto Tecnológico de Aeronáutica, São José dos Campos, São Paulo, Brazil*
[b]*Jet Propulsion Laboratory - NASA/Caltech, Pasadena, California, USA*
[c]*University of California Santa Cruz, Santa Cruz, California, USA*

Abstract

Controlling congestion is critical to ensure adequate network operation and performance. That is especially the case in networks operating in challenged- or extreme environments where episodic connectivity is part of the network's normal operation. Consequently, the "pure" end-to-end congestion control model employed by the Internet is not adequate. Our goal is to study congestion control mechanisms that have been proposed for these so-called disruption tolerant networks, or DTNs. In this paper, we conduct a performance study comparing existing DTN congestion control mechanisms for two main application domains, namely: inter-planetary (IPN) and terrestrial networking applications. Our results confirm that congestion control helps increase message delivery ratio, even in highly congested network scenarios. Furthermore, the results show that existing DTN congestion control mechanisms do not perform well in IPN scenarios. Our study also suggests that good design principles for congestion control in DTN scenarios include: combining reactive- and proactive control, using local information instead of global knowledge, and employing mechanisms that are routing protocol independent. One important conclusion from our quantitative study is that there is currently no universal congestion control mechanism that fits all DTN scenarios and applications.

[*] Corresponding author
Email addresses: aloizio@ita.br (Aloizio P. Silva), scott.c.burleigh@jpl.nasa.gov (Scott Burleigh), hirata@ita.br (Celso M. Hirata), katia@soe.ucsc.edu (Katia Obraczka)

Keywords: Delay and Disruption Tolerant Networks, Interplanetary Networks, Congestion Control, Network Performance
2015 MSC: 00-01, 99-00

1. Introduction

In the last 10 years, applications such as environmental sensing, habitat monitoring, emergency response, disaster recovery, and bridging the digital divide, to name a few, have raised great interest in so-called challenged network environments. In such environments, also known as delay and disruption tolerant networks, or DTNs, continuous end-to-end connectivity cannot be guaranteed and the communication channel may be subject to arbitrarily long signal propagation delays, lapses in connectivity, and high error rates. Under these conditions, participating nodes must store in persistent storage data they are transmitting or forwarding until a contact opportunity arises, i.e., until the node has a suitable next-hop neighbor that can receive the data.

Congestion control in challenged environments is thus critical to ensure nodes are congestion-free and can serve as relays when needed so messages can be delivered end-to-end. Because DTNs violate the fundamental assumptions underlying the TCP/IP (Transmission Control Protocol/Internet Protocol) protocol architecture, namely the existence of an end-to-end path between nodes and short delays, they cannot employ the Internet's congestion control principles. The technical challenges posed by DTN congestion control combined with its impact on performance motivated a number of research efforts aiming at developing novel congestion control schemes for DTNs [9, 15, 22, 29, 19].

Our goal with this study is to understand the performance trade-offs raised by existing DTN congestion control mechanisms and how they behave in a wide range of DTN scenarios. We focus on two types of DTN applications, namely deep space communications, also known as Interplanetary Internet (IPN) [6] and terrestrial applications [1]. Terrestrial DTNs can target a wide range of applications including sensor networks

[1] A preliminary version of this work focusing on IPN environments is described in [24]

for environmental- and habitat monitoring, vehicular networking applications, law enforcement and first responder services, to name a few. Our study compares the performance of different congestion control schemes in both IPN and terrestrial scenarios. To our knowledge, this is the first performance study on DTN congestion control examining both interplanetary and terrestrial applications, thus exploring the feasibility of a universal congestion control mechanism.

We also explore the impact of different routing protocols and node mobility models and evaluate the different congestion control strategies using such performance metrics as average delivery ratio, average latency, and overhead. Another goal of this work is to provide insight on good design principles for congestion control schemes so that they can be applied to a variety of DTN scenarios.

The remainder of the paper is organized as follows. In Section 2, we provide an overview of DTN and IPN environments and Section 3 describes the congestion control schemes we studied. Section 4 describes our experimental methodology while Sections 5 and 6 present the results of our comparative study. Section 7 provides a discussion of the results and Section 8 concludes the paper.

2. Background

2.1. Interplanetary Internetworking

As described in [2, 3, 6], an Interplanetary Internet includes the IPN Backbone, IPN External Networks, and Planetary Networks (PNs). The IPN Backbone makes possible the communication among the Earth, other planets, space probes, and spacecraft through satellites. The IPN External Network includes, for instance, spacecraft flying in deep space between planets, space probes, and orbiting space stations. A PN is composed of the PN's Satellite Network and the PN's Surface Network. The former includes links among surface nodes, orbiting satellites, and IPN Backbone Nodes, providing relay services between surface networks and the backbone as well as between two or more parts of the surface network. Surface networks provide communication between surface elements, such as rovers and sensor nodes.

The main challenges affecting IPNs can be summarized as follows [2, 8, 30, 32, 28]:

- **Intermittent connectivity:** disconnections can happen due to planetary motion as well as the movement of celestial bodies, spacecrafts, rovers, etc.
- **Long and variable delays:** the deep space connection may have extremely high round-trip latencies caused by astronomical distances, e.g., the round-trip time (RTT) for radio communication from Mars to Earth can take between 3 minutes minimum to 30 minutes maximum. This happens because the distance between the Earth and Mars varies enormously depending on their relative positions in their orbits around the Sun.
- **High bit error rates:** the uncorrected bit error rate (BER) for deep space radio communication is high (around 10^{-1}) due to extreme environment conditions (i.e., cosmic radiation leads to signal corruption). Strong forward error correction coding is applied to reduce the observed BER to a rate that is on the order of 10^{-5} to 10^{-6}, still far higher than in communications over optical fiber in the Internet.
- **Asymmetric data rates:** the asymmetry in date rates on space links is typically of the order of 1:1000 or higher. Communications channels between spacecraft and the ground are frequently asymmetric in terms of both channel capacity and error characteristics. This asymmetry is a result of various engineering tradeoffs (such as power, mass, and volume), as well as the fact that for scientific missions, most of the data originates at the satellite and flows to the ground. The return link is generally used for commanding the spacecraft, not bulk data transfer [10].

2.2. Terrestrial DTN

Terrestrial DTNs find a variety of applications including (wireless) sensor networks (e.g., for environmental- and habitat monitoring often times deployed in remote regions possibly under extreme conditions), vehicular networks, emergency rescue and disaster relief operations. Similar to deep space networks, some of the main challenges affecting terrestrial networks can be summarized as follows:

- **Intermittent connectivity**
- **Arbitrarily long and highly variable delays**
- **Highly variable bit error rates:** the BER varies according to the environment (e.g., deep-sea sensor networks or military/civilian submarine communication). For example, in a wireless sensor network the BER may be on the order of 10^{-1} to 10^{-3} [21][11] but is frequently lower than in the deep space environments.
- **Data rates:** Data rates are frequently symmetric but may be asymmetric in some environments (e.g., underwater communications).

However, unlike IPNs where contacts are typically governed by planetary movement [12], in terrestrial DTNs contacts are not usually scheduled. They are often random also known as “opportunistic”.

3. Selected DTN Congestion Control Mechanisms

For our comparative study, we picked a subset of DTN congestion control mechanisms from the schemes presented in our DTN congestion control survey [9] that we consider representative of the current DTN congestion control state-of-the-art. More specifically, we use the taxonomy we proposed in our survey to select mechanisms that utilize different types of control strategies. For example, we evaluate protocols that use reactive, proactive, or hybrid (i.e., reactive and proactive) control; we also consider open- versus closed-loop control, as techniques that are routing protocol dependent or independent, as well as different congestion detection mechanisms. Table 1 summarizes the distinguishing features of the selected DTN congestion control protocols.

One of our main goals is to explore how current DTN congestion control protocols behave in different DTN scenarios, including the Interplanetary Internets (IPNs). As we previously pointed out, to-date, no DTN congestion control scheme has been designed for deep space communication applications. The DTN congestion control schemes we consider in our performance study are: RRCC (Retiring Replicants Congestion Control) [26], AFNER (Average Forwarding Number based on Epidemic Routing) [31], SR (Storage Routing) [23] and CCC (Credit-based Congestion Control) [18].

Below, we briefly describe each of these mechanisms as follows: for each scheme, we specify the control strategy adopted and the steps performed to avoid or mitigate congestion. Our survey [9] provides a more detailed description of the DTN congestion control protocols studied as well as other DTN congestion control mechanisms that have been proposed to-date. Table 1 summarizes the four mechanisms based on the taxonomy presented in [9]. The column *Routing* indicates whether there is any dependency relationship between congestion control and routing. Congestion control approaches that can work with any routing mechanism are said to be routing-protocol independent; those that cannot are said to be routing-protocol dependent.

Table 1: Classification of selected congestion control mechanisms according to [9]

Mechanism	Congestion Detection	Proactive or Reactive Control	Routing	Evaluation Platform
RRCC [26]	Drop rate	H	D	ONE
AFNER [31]	Network capacity	H	D	NS-2
SR [23]	Buffer availability	R	D	DTNSim
CCC [18]	Buffer availability	H	I	ONE

H: Hybrid R: Reactive D: Dependent I: Independent

3.1. Retiring replicants congestion control (RRCC)

Retiring Replicants Congestion Control (RRCC) [26] aims at reacting to DTN congestion by controlling the number of copies a node forwards for each message, i.e., the *replication degree*, during an encounter. Nodes employ local knowledge to approximate global network behavior and adjust the node's replication degree to control congestion and maximize delivery. According to [26], RRCC's congestion control is independent from the routing protocol and does not interfere with forwarding decisions. However, we consider RRCC routing-protocol dependent since, as shown by our results, RRCC's performance depends on the underlying routing protocol. This is consistent with the classification we present in our DTN congestion control survey [9], which places RRCC in the routing-protocol dependent category.

To adjust its replication degree, a node must observe the current level of congestion, the *congestion value* or CV, in the network. The congestion value is computed as the ratio of drops over the number of message copies (or the number of replicated messages). Whenever the congestion value is updated, the node also adjusts the number of message copies ($limit$ that refers to the mumber of message copies a node is allowed to create for forwarding) according to an additive increase, multiplicative decrease (AIMD) algorithm. The pseudo-code for the RRCC algorithm is shown in Algorithm 1. For our study, we have implemented RRCC taking into account an approximation of global metrics (e.g., drops and number of message copies) as suggested by the authors. In the worst case, this algorithm has a cost $O(n)$, where n is the number of exchanged messages during the encounter.

Algorithm 1 Pseudo-code of RRCC Congestion Control Mechanism

```
 1: procedure RRCC
 2:     drops ← 0;
 3:     reps ← 0;
 4:     limit ← 0;
 5:     CV ← Double.MaxValue;
 6:     $A_i$ ← 1;                      ▷ the replication limit could be increased by a fixed amount
 7:     MD ← 0.2;                       ▷ multiplicative factor to congestion value
 8:     if event.equal(DROP) then
 9:         drops ← $drops + 1$;
10:     else if event.equal(RECEIVE_MESSAGE) then
11:         reps ← reps + 1;
12:     else if event.equal(CONTACT) then                        ▷ contact node b
13:         $hopsCount$ ← $message.getHopsCount - 1$;   ▷ № of hops this message has passed
14:         d ← drops + b.drops;
15:         r ← $reps + b.reps + hopsCount$;
16:         reset(drops, reps);
17:         $CV^{'} \leftarrow \beta \times (d/r) + (1 - \beta) \times CV$;
18:         if $CV^{'} \leq CV$ then
19:             limit ← limit + $A_i$;
20:         else
21:             limit ← limit × MD;
22:         end if
23:         CV ← $CV^{'}$;
24:     end if
25: end procedure
```

The congestion value is calculated by Equation 1, where β is a weight assigned to CV sample. Its value is set in 0.9. According to the algorithm, if the current congestion value $CV^{'}$ is higher than the previous value CV, there is growing congestion and the limit is reduced by a multiplicative factor *MD* $= 0.2$ whereas when congestion is decreasing the limit is increased by $A_i = 1$ (see this in Algorithm 1 lines 19 and 21). In RRCC, when a node wishes to drop a message because of buffer overflow, it adopts a random drop policy.

RRCC uses an adaptive replication management strategy based on the estimates of global network information, which is challenging to predict with accuracy. However, RRCC adopts a hybrid congestion control approach combining reactive and proactive control, which tends to deliver better responsiveness to congestion.

$$CV^{'} = \beta CV_{sample} + (1 - \beta)CV \tag{1}$$

3.2. Average forwarding number based on epidemic routing (AFNER)

The congestion control strategy called "average forwarding number based on epidemic routing" (AFNER) is described in [31] and works as follows. When a node's storage is full and the node needs to accept another incoming message, the node randomly drops one of the messages whose *forwarding number* is larger than the network's average forwarding number. The forwarding number of a message is defined as the number of copies of that message, while the average forwarding number is the mean forwarding number of all the messages currently in the network. AFNER assumes that messages are not fragmented and are transmitted in FIFO (First-In First-Out) order from one node to another during the contact period. AFNER is based on two principles:

- First, messages in a node's buffer are sorted in ascending order according to their forwarding number.
- Second, a message will be delivered at ultimate destination if and only if its forwarding number is greater than or equal to the average forwarding number.

The pseudo-code of the AFNER algorithm is shown in Algorithm 2, which has been implemented as part of this quantitative study. AFNER's computational complexity is given by is $O(n\ log(n)) + m)$ in the worst case, where n is the node's buffer size and m the total number of messages in network.

Some of AFNER's features deserve to be highlighted. Notably, AFNER's congestion control depends on the average forwarding number which is not easily computable in a real DTN since it requires global knowledge. In addition, as its name implies, AFNER relies on epidemic routing, which incurs high overhead and may congest the network; it also adopts a reactive congestion control approach.

Algorithm 2 Pseudo-code of AFNER Congestion Control Mechanism

```
procedure AFNER
    $F_{num}$;    ▷ the number of message in the buffer of congested node.
    $F_t$;    ▷ the length of the existing messages in the buffer at congested node.
    $SV_a$;    ▷ the summary vector of node a.
    $SV_b$;    ▷ the summary vector of node b.
    $F_{ave}$;    ▷ the average forwarding number of all of messages.
    if node a encounter node b then
        both nodes share their summary vector ($SV_a$ e $SV_b$);
    end if
    if currentQueueSize $\geq$ messageSize then
        congestion does not occur, the message can be stored;
        $return$;
    else    ▷ congestion has occurred.
        $F_t = F_{num} \times 1$;    ▷ congested node computes $F_t$.
    end if
    if (messageSize $-$ currentQueueSize) $< F_t$ then
        NumMsgDeleted $\leftarrow 0$;
        while $NumMsgDeleted \neq$ (messageSize $-$ currentQueueSize) do
            delete message;    ▷ according to descending order of the forwarding number.
            NumMsgDeleted $\leftarrow$ NumMsgDeleted $+ 1$;
        end while
    else if (messageSize $-$ currentQueueSize) $\geq F_t$ then
        delete all messages whose forwarding number is bigger than $F_{ave}$
    end if
end procedure
```

3.3. Storage routing (SR)

Storage Routing (SR) employs a strategy based on migrating excess messages to alternate storage locations, i.e., neighboring nodes with available storage capacity, during congestion [23]. When nodes that were previously at risk of congestion manage to reduce their buffer occupancy, messages that were migrated are retrieved. SR operates as a local routing protocol diverting messages from their conventional routing path for later forwarding.

SR is invoked by a DTN node when a message arrives at the node that is nearing congestion (see Algorithm 3 Line 4). In other words, SR detects congestion by verifying if the arriving message's size is greater than the node's currently available buffer space. If this is the case, the node is considered to be congested, and SR determines a set of messages to migrate to a set of selected neighbors. Nodes selected as migration targets are called *alternate custodians*. When the risk of congestion subsides, SR invokes a retrieval selection algorithm to determine which nodes it will contact with custody requests for previously migrated messages. If the alternate custodians no longer contain the migrated message, a NACK (negative acknowledge) message is returned; otherwise the migrated message is returned.

SR can be summarized as follows (see Algorithm 3 for SR's pseudo-code):

Message selection chooses messages to be "pushed" to neighbor nodes (instead of simply discarding them, which is the case in traditional drop-based buffer management). Messages are deleted only when there is no available storage at any neighboring node.

Node selection produces a set of nodes to which messages can be migrated. The algorithm selects nodes based on an aggregate migration cost metric ($C_{c,v}(l)$) calculated by Equation 2.

$$C_{c,v}(l) = T_{c,v}(l)w_t + S_v(l)w_s \tag{2}$$

where migration cost from custodian node c to a neighborhood node v of a message with length l is the weighted summation of the storage cost $S_v(l)$ and transmission cost $T_{c,v}(l)$. The values of w_t and w_s are set according to the importance of transmission cost versus storage cost. Transmission cost is a function of the latency $L_{c,v}$ and bandwidth $B_{c,v}$ on the path node $c \to v$ and the message size l. Thus, $T_{c,v}(l)$ can be obtained by equation 3.

$$T_{c,v}(l) = log((L_{c,v} + (l/B_{c,v}))/(10^{-6}))/10 \tag{3}$$

The storage cost is a function of the available storage for migration to node v and is given by Equation 4, where A_v is the available storage and Max_v is the maximum node storage.

$$S_v(i) = \begin{cases} A_v \div Max_v & \text{if} \quad l \leq A_v \\ +\infty & \text{if} \quad l > A_v \end{cases} \tag{4}$$

Taking into account the node selection strategy, SR's worst case cost is $O(n) + k$, where n is the buffer size and k is the number of alternate nodes. SR does not work well

Algorithm 3 Pseudo-code of SR Congestion Control Mechanism

```
1: procedure SR
2:     *bufferSize*;                                  ▷ free buffer space.
3:     $S \leftarrow 0$;                              ▷ counter of the amount of available storage by migrating messages.
4:     if *m.size()* > *bufferSize* then
5:         $R \leftarrow m.size() - bufferSize$;
6:         $Mp = n.MessageSelection()$;
7:         $x = n.NodeSelection()$;
8:         if $x \neq null$ then
9:             $n.sendMessage(Mp, x)$;
10:        else
11:            $n.deleteMessage(Mp)$;
12:        end if
13:        $S = Mp.size() + S$;
14:        if $S \geq R$ then
15:            *enqueue m at n*;
16:        else
17:            go to 7
18:        end if
19:    end if
20: end procedure
```

in the case of sparse networks where neighbors are not always available. Furthermore, there are not guarantees that there will exist neighbor nodes with available buffer space when needed.

3.4. Credit-based congestion control (CCC)

A heuristic-based congestion control mechanisms called credit-based congestion control (CCC) that handles congestion proactively and reactively is presented in [18]. CCC was motivated by the so-called "looping problem" that happens when a node that has removed a message from its buffer continues to receive the message from other nodes. To address this problem, CCC proposes a looping control scheme and a congestion control policy. In order to yield high delivery ratio with low number of replicas, CCC tries to delete messages when congestion builds up at a node (reactive approach). Messages become eligible to be dropped when deemed obsolete according to their *time-dependent credit* or if they have been forwarded too many times (proactive approach). A refilling and refunding technique is applied to the message's time-dependent credit when node pairs encounter and exchange messages.

CCC works as follows: when a message has been generated, it is assigned the maximum amount of credit. As time passes, one credit is decreased at every time unit. When two nodes encounter each other, they exchange messages, and for each message exchanged, the sender deducts a *penalty* value from the copy of the message it keeps in its buffer. The receiver in turn adds a *reward* value to the credit of the copy it just received. Our implementation of CCC uses Equations 5 and 6 to compute functions for refilling and refunding, respectively.

$$\begin{aligned} C_{sender}(t+1) = max(C_{sender}(t) - \ penalty, \\ minimum - value) \end{aligned} \tag{5}$$

$$\begin{aligned} C_{receiver}(t+1) = max(C_{sender}(t) + \ reward, \\ initial - value) \end{aligned} \tag{6}$$

where C_{sender} is the message's credit at the sender, $C_{receiver}$ is the credit of replicated message at the receiver, before and after the encounter. Operations executed by this mechanism, such as refilling and refunding, do take time, but this time is constant. In the worst case, CCC's cost is $O(n)$, where n is the number of exchanged messages during the encounter. Note that this technique might not work well if node encounters are not frequent enough.

4. Experimental Methodology

We conducted experiments using the ONE (Opportunistic NEtwork) Simulator platform [17], which is a discrete event simulator specific for DTN environments. We should point out that the ONE simulator does not implement transport-level protocols. As such, the congestion control mechanisms under study were implemented "side-by-side" with routing/forwarding. For example, RRCC uses its notion of congestion level to adjust the degree of replication used when messages are forwarded.

4.1. DTN Scenarios

In order to study the performance of the four different congestion control mechanisms described in Section 3, we simulate two main DTN application domains, namely deep space communications (or IPN) and terrestrial applications. The parameters of the ONE simulator and their values for the IPN and terrestrial scenarios are listed in

Tables 3 and 4, respectively. We discuss our choice of values for the simulation parameters in Sections 4.1.1 and 4.1.2 below.

A scenario without congestion control is considered as baseline for our comparative study. Another interesting aspect of our study is to understand the dependence of the congestion control schemes on the underlying routing protocol [14]. To this end, we use three different routing protocols, namely Epidemic [27], ProPHET (Probabilistic Routing Protocol) [20], and Spray and Wait [25], each of which is briefly described below. It is important to note that like the congestion control mechanisms we study, these routing protocols were not designed to operate in deep space communication. They were designed for terrestrial networking.

- **Spray-and-Wait**: A node sprays copies of messages, such that, the number of message copies is limited to a configurable maximum. In our experiments we set the number of copies to 10.
- **ProPHET**: When a node A encounters a node B, it decides whether to pass to B a copy of a message destined to C based on B's encounter history with C.
- **Epidemic**: A node delivers copies of its messages to all encountered nodes.

4.1.1. Interplanetary Network (IPN) Scenario

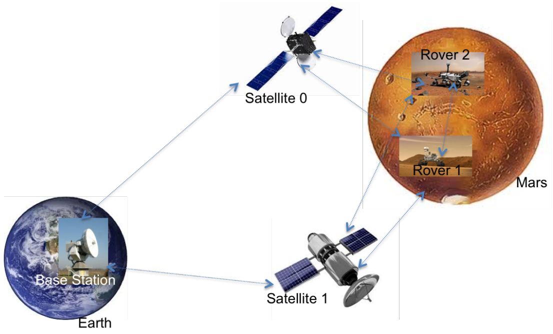


Figure 1: Interplanetary network scenario.

Our IPN (Interplanetary Network) scenario features high latency (because of astronomical distances) and scheduled contacts, i.e., node encounters that are known a priori. There are five nodes, representing a Base Station on the surface of the Earth that sends data to two rovers on Mars through two satellites located near Mars (see Figure 1). Simulation parameters are set to correspond to realistic conditions. Their values were selected based on discussions with researchers from the Interplanetary Network Laboratory at JPL-NASA and documents published by the DTNRG [1]. For instance, links between the satellites and the rovers on the Martian surface are set to $1s$ propagation delay while the links between the base station on Earth and the satellites in Mars are set to 240 s propagation delay.

Table 2: Example of scheduled contact table.

Time (s)	Identifier	Initial Node	End Node	State
2000	CONN	Base Station	Satellite 0	up
3000	CONN	Base Station	Satellite 0	down
6000	CONN	Base Station	Satellite 1	up
7000	CONN	Base Station	Satellite 1	down
10000	CONN	Base Station	Satellite 0	up
11000	CONN	Base Station	Satellite 0	down
17000	CONN	Base Station	Satellite 1	up
18000	CONN	Base Station	Satellite 1	down
20000	CONN	Rover 2	Rover 1	up
21000	CONN	Rover 2	Rover 1	down
20000	CONN	Satellite 1	Rover 2	up
21000	CONN	Satellite 1	Rover 2	down
21100	CONN	Satellite 1	Rover 1	up
22100	CONN	Satellite 1	Rover 1	down
25000	CONN	Base Station	Satellite 1	up
26000	CONN	Base Station	Satellite 1	down
30000	CONN	Satellite 0	Rover 2	up
31000	CONN	Satellite 0	Rover 2	down
33000	CONN	Satellite 1	Rover 1	up
34000	CONN	Satellite 1	Rover 1	down
38000	CONN	Satellite 0	Rover 2	up
39000	CONN	Satellite 0	Rover 2	down

We mainly focus on testing how congestion is affected by the inter-contact time and the duration of contact. Thus a scheduled contact table was created (see Table 2) which records the time when a connection is created from one node to another node. Also, it records the time when a connection drops between two nodes. The Up and Down connection times are asserted a priori in the scheduled contact table. Note that the communication channel is asymmetric. Although Table 2 does not indicate when the connection from *Satellite 0* to *Base Station* is Up and Down, we assume that the state of the connection from *Satellite 0* to *Base Station* is always the same as the state of the connection from *Base Station* to *Satellite 0*. For example, according to the first line of Table 2, at time 2000 s, the connection between *Base Station* and *Satellite 0* is “up” but goes“down” at time 3000 s. Then, the same nodes are again in contact at time 10000 s (line 5) for 1000 s (line 6).

We use 5 different scheduled contact tables that differ in inter-contact time. We named the five different contact schedules as Contact1, Contact2, Contact3, Contact4, and Contact5. We increase the inter-contact time by 1000 s as we go from Contact$i$ to Contact$i + 1$.

Table 3: Simulation parameters and their values for the IPN scenario

Parameters		
Name	Description	Value
Scenario.endTime	simulation time	$43200\,\mathrm{seconds}$
btInterface.transmitSpeed	bandwidth	$2.5\,\mathrm{Mbps}$
btInterface.transmitRange	transmitting range	$150\,\mathrm{m}$
Group.router	routing protocol	[EpidemicRouter, ProphetRouter, SprayAndWait-Router (10 msg copies)]
Group.movementModel	mobility model	StationaryMovement
Group.bufferSize	node buffer size	$[1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000]\,\mathrm{kbytes}$
Group.msgTTL	message time to live	$43200\,\mathrm{seconds}$
Group.nrofHosts	number of nodes in network	5
Movimentmodel.worldSize	area where simulation takes place	$6\,\mathrm{km} \times 6\,\mathrm{km}$
Events1.size	message size	$\{50, 100\}\,\mathrm{KB}$
Events1.interval	i.e. one new message every 1 to 100 seconds	[1-100, 1-200, 1-300, 1-400, 1-500] seconds

Nodes generate messages according to the $Events1.interval$ parameter (see Table 3) where its value is $[x_i, x_j) = x_i \leq$ *next-event-time* $< x_j$. Uniformly distributed value between x_i (inclusive) and the x_j value (exclusive). We vary the message generation rate according to the values listed in Table 3 to show how this parameter affects the performance of the different congestion control schemes.

4.1.2. Terrestrial Scenario

The terrestrial scenarios we simulate include 50 nodes that move according to a predefined mobility regime (as discussed in more detail below). As they move, nodes encounter one another opportunistically, and during these “opportunistic contacts”, nodes can exchange messages. Figure 2 shows the output of the ONE simulator’s graphic interface illustrating a terrestrial DTN scenario we simulated. Table 4 lists the simulation parameter settings we used in our experiments and their values. We assume that all nodes have the same transmission range. Node transmission range as well as the values of the other simulation parameters were set based on experiments reported in the original papers presenting the congestion control mechanisms we studied. The scenarios we simulate can correspond for example to a wireless network of mobile sensing nodes performing surveillance or situational awareness.

Table 4: Simulation parameters and their values for terrestrial scenario

Parameters		
Name	Description	Value
Scenario.endTime	simulation time	43200 seconds
btInterface.transmitSpeed	bandwidth	2.5 Mbps
btInterface.transmitRange	transmitting range	150 m
Group.router	routing protocol	[EpidemicRouter, ProphetRouter, SprayAndWaitRouter (10 msg copies)]
Group.movementModel	mobility model	[RandomWayPoint, RandomWalk, ShortestPathMapBasedMovement]
Group.bufferSize	node buffer size	1000 kbytes
Group.msgTTL	message time to live	43200 seconds
Group.nrofHosts	number of nodes in network	50
Group.speed	max and min speed that the nodes must move	$\{0.5, 1.5\}$ m/s
Movimentmodel.worldSize	area where simulation takes place	1 km $\times$ 1 km (RandomWayPoint, RandomWalk) and 6 km$x6$ km (ShortestPathMapBasedMovement)
Events1.size	message size	$\{50, 100\}$ KB
Events1.interval	i.e. one new message every 1 to 100 seconds	[1-100, 1-200, 1-300, 1-400, 1-500] seconds

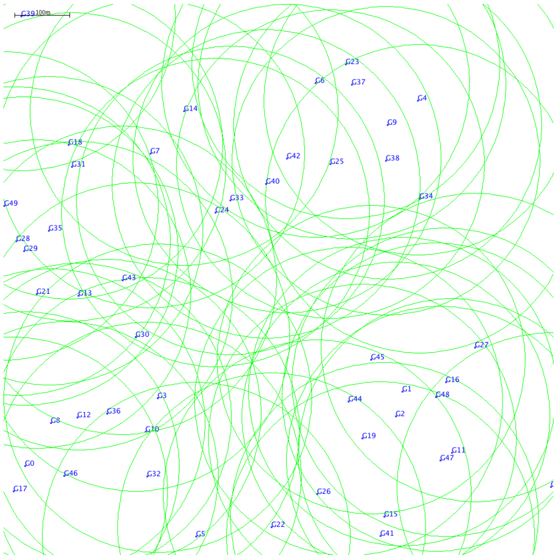


Figure 2: Snapshot of simulated terrestrial DTN scenario.

Three mobility models were used, namely Random Walk (RW), Random Way Point (RWP), and Shortest Path Map-Based Movement (SPMBM). In RW [7], a node randomly chooses a destination within the simulation area. It then moves from its current location to the new one with speed uniformly distributed within the interval given by $Group.speed$. When it arrives at its destination, it picks another one and repeats the steps above. The RWP mobility model [7] is a generalization of RW and works as follows: a mobile node picks a random destination within the simulated area; it then moves to that destination with constant speed chosen as a uniformly distributed random number in the interval $Group.speed$. When the node reaches its destination, it pauses for some time. In our simulations, the pause time is a uniformly distributed random number between $[0, 120]$ seconds. After that, the node picks another random destination and repeats the steps above. The SPMBM [16] model uses Dijkstra's shortest path algorithm to calculate the shortest path from the current location to a randomly selected destination. Similarly to RWP, when the node arrives at its destination, it also uses a uniformly random pause time between $[0, 120]$ seconds.

4.2. Performance Metrics

We consider three main performance metrics, namely:

1. **Delivery ratio** is the ratio between the number of received messages at destination nodes to the number of created messages (see Equation 7).

$$\text{delivery ratio} = \frac{\text{number of received messages}}{\text{number of created messages}} \times 100\% \tag{7}$$

2. **End-to-End latency** is the average time interval to deliver messages to their destinations (see Equation 8 where t_i is defined as the time that message i took to reach the destination and t_c is the message creation time).

$$\text{end-to-end latency} = \frac{\sum_{i=1}^{\text{number of messages received}} (t_i - t_c)}{\text{number of messages received}} \tag{8}$$

3. **Overhead** is the difference between the number of delivered messages and the number of relayed messages, divided by the number of delivered messages (see Equation 9). The overhead reflects the number of messages relayed to deliver a single message. The lower the value of overhead, the more efficient the strategy is.

$$\text{overhead} = \frac{\text{number of relayed messages} - \text{number of received messages}}{\text{number of received messages}} \tag{9}$$

In our results shown in the remaining of this section, we use 95% confidence intervals and each data point presented in the graphs below is computed as the average over 5 runs. Note that when graphs do not show the standard deviation, it is because the value is either zero or approximately zero.

5. Simulation Results for IPN Scenario

In this section, we present simulation results from our IPN experiments. To evaluate the impact of congestion control, we try to generate enough load to congest the network. The goal of the evaluation is to show the performance of each congestion control mechanism relative to the baseline scenario, i.e., no congestion control, in terms of delivery ratio and message delivery latency. We also consider the impact of the underlying routing protocol; to this end, except for AFNER which uses Epidemic routing, we run each congestion control strategy with different routing protocols, i.e., Epidemic, ProPHET, and Spray and Wait. Tables 2 and 3 list the simulation parameter settings.

Figure 3 shows message delivery ratio as a function of the message generation period. As expected, for longer message generation periods, i.e., lower message generation frequencies, the delivery ratio increases. However, delivery ratios are overall quite low since, at this level of congestion, newly generated messages cause older messages to be discarded before they have time to be delivered. This result confirms that existing congestion control protocols perform quite poorly in IPN scenarios. In fact, in some cases, the baseline scenario, i.e., where no congestion control is used, shows higher delivery ratios than scenarios where congestion control is used. That is often the case for AFNER for mainly two reasons: its use of epidemic routing and reactive control; the fact that AFNER's congestion control approach relies on global knowledge also makes it less effective.

5.1. Different Inter-Contact Times

The results presented in this section allow us to analyze the impact on congestion control mechanisms when varying the inter-contact time. Table 5 shows an example of the configuration of inter-contact time for one pair of nodes, namely *Base Station* and *Satellite 1*, where column *Inter-Contact Time* depicts the inter-contact times which are managed by *up* connection and its time (columns *State* and *Time(s)*, respectively).

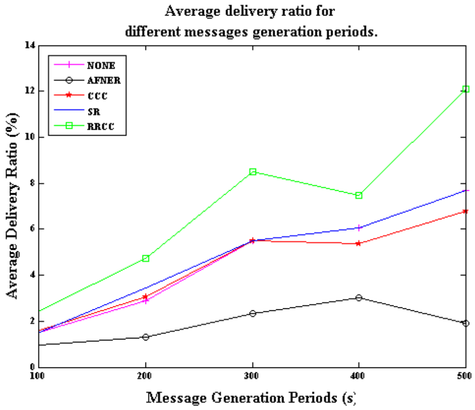


Figure 3: Message delivery ratio for IPN Scenario as a function of the message generation period (epidemic routing, buffer size of 4000 kbytes, transmit speed of 2.5 Mbps, contact duration 1000 s, and inter-contact time 1000 s).

Table 5: Inter-contact times for the encounters between *Base Station* and *Satellite 1*.

Inter-Contact Time	State	Time(s)	Initial Node	End Node
1000	up	1000	Base Station	Satellite 1
	down	2000	Base Station	Satellite 1
2000	up	2000	Base Station	Satellite 1
	down	3000	Base Station	Satellite 1
3000	up	3000	Base Station	Satellite 1
	down	4000	Base Station	Satellite 1
4000	up	4000	Base Station	Satellite 1
	down	5000	Base Station	Satellite 1
5000	up	5000	Base Station	Satellite 1
	down	6000	Base Station	Satellite 1

Figure 4 illustrates the concepts of inter-contact time and contact duration through an example involving the *Base Station* and *Satellite* nodes. Consider that the *Base Station* and the*Satellite* only encounter once during a normal day. In this example in *Day 1* at time 1000 s the *Base Station* connects with *Satellite 1* (start of the contact). They remain in contact/ connected until 2000 s (end of the contact). In this case, the contact duration was 1000 s (see the solid horizontal line). In the *Day 5* they connect with each other at 5000 s (start of the contact) and end the connection at 6000 s (end of the contact). Note that the inter-contact time between the *Base Station* and *Satellite 1* increases for different days (see the dashed line).

Figure 5 shows the average delivery ratio for different inter-contact times, where the slim bars refer to delivery ratio values when no congestion control is used. As expected, we observe that, for longer inter-contact times, the delivery ratio decreases.

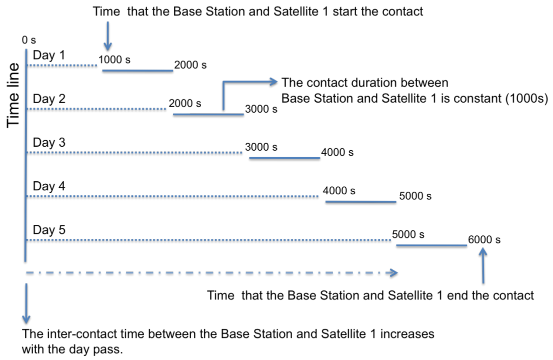
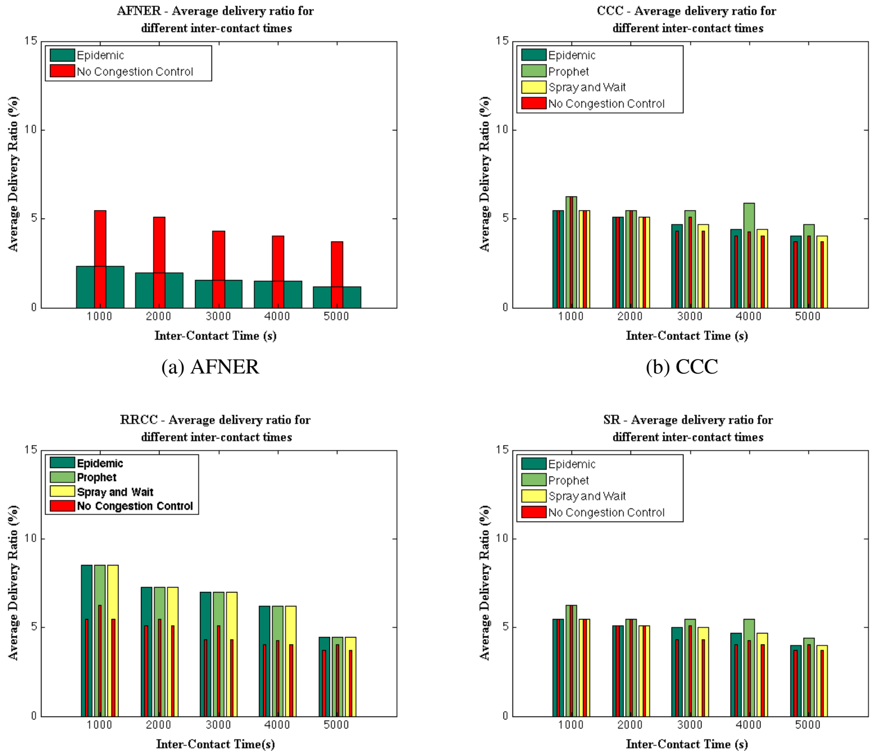

Time that the Base Station and Satellite 1 start the contact
0 s
Time line
Day 1
1000 s
2000 s
The contact duration between
Base Station and Satellite 1 is constant (1000s)
Day 2
2000 s
3000 s
Day 3
3000 s
4000 s
Day 4
4000 s
5000 s
Day 5
5000 s
6000 s
Time that the Base Station and Satellite 1 end the contact
The inter-contact time between the Base Station and Satellite 1 increases
with the day pass.

Figure 4: Inter-contact time and contact duration for the encounters between *Base Station* and *Satellite 1*.

This happens because of the lower number of transmission opportunities, which results in longer data queues and consequently higher probability of data being dropped as node buffers fill up. As a result, the data delivery ratio decreases.

We observe that CCC's performance in terms of delivery ratio (Figure 5b) is quite similar to SR's (Figure 5d). CCC's and SR's delivery ratios see a slight increase when ProPHET is used as the underlying routing protocol. Our hypothesis is that, since ProPHET bases its routing decisions on past contact history, it benefits from scenarios where contacts are repetitive; this is often the case in IPN environments, where node contacts typically follow a well-known schedule. Epidemic and Spray and Wait routing fit opportunistic DTNs better. RRCC's dependence on the routing protocol is less noticeable in the IPN scenarios than in the terrestrial ones (as shown by our results for terrestrial scenarios in Section 6 below).

It is worth pointing out that, except for AFNER, as inter-contact times increase, the congestion control mechanisms yield higher delivery ratios when compared to the configuration that lacks congestion control altogether. As previously observed, AFNER's deliver ratios are lower than when no congestion control is used. Besides using epidemic routing which contributes to congestion, AFNER uses the network's average forwarding number to mitigate congestion. Thus when the inter-contact times increase, more messages are waiting to be forwarded. As a result, buffers fill up and AFNER starts to discard messages based on the average forwarding number. This leads to lower delivery ratios. Note that for longer inter-contact times the nodes' buffer can fill up be-



(a) AFNER

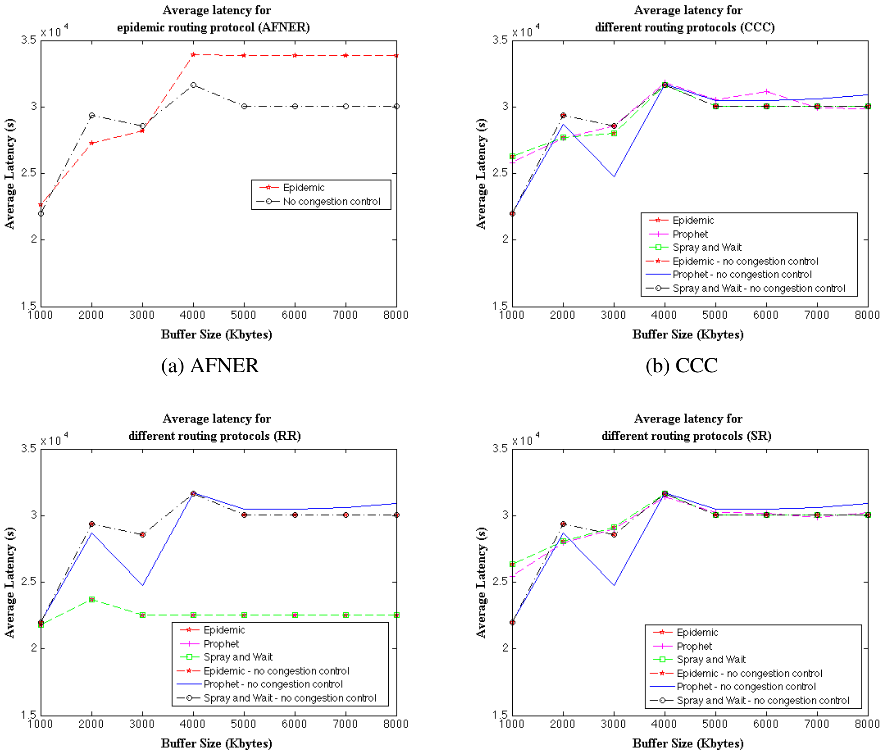
(b) CCC

(c) RRCC

(d) SR

Figure 5: Average delivery ratio for different inter-contact times (IPN scenario, transmit speed of 2.5 Mbps, buffer size of 4000 kbytes, contact duration 1000 s, inter-contact time 1000 s, and message generation period of 300 s; slim bars represent the network without any congestion control mechanism).

fore the contact begins. This happens because the nodes are able to generate messages according to the message generation ratio. Essentially, delivery and forwarding happen only during a contact event. Therefore, the probability that a node's buffer is full increases as inter-contact times increase. When congestion builds up, RRCC adjusts replication levels accordingly which increases delivery ratio when compared with the other mechanisms. CCC's congestion control is based on discarding older messages. Consequently, obsolete messages and those that have been forwarded many times are dropped first. This approach helps to improve the delivery ratio when increasing inter-contact time. While SR's message migration mechanism yields comparable delivery ratios to CCC, it may lead to high delays and does not work if the node does not have any neighbors or their buffers are full. In the latter case, SR starts to discard messages.



(a) AFNER

(b) CCC

(c) RRCC

(d) SR

Figure 6: Average latency for different buffer sizes (IPN scenario, transmit speed of 2.5 Mbps, inter-contact time 1000 s, contact duration 1000 s, and message generation period of 300 s).

Figure 6 shows the average message delivery latency of the congestion control mechanisms under study for different routing protocols. One interesting observation is that the minimum latency value is approximately 2.3×10^4 s, that is, a hundred times greater than the propagation delay for the links between the base station on Earth and the satellites at Mars which were set up to 240 s. We argue that this happens due to the longer inter-contact times and the congestion effect. The IPN environment is subject to high latency and the contacts between nodes are normally scheduled or probabilistic. In this case, for the modeled scenario, the latency is not expected to decrease linearly with increasing buffer size, but rather to remain unchanged or decrease with some fluctuations. This can be explained by the fact that when buffer size is increased the message has a higher probability of being delivered, because the chance of the message being discarded due to buffer overflow is reduced. Thus, depending on the arrival time in the buffer and the time of the next scheduled contact, the message can remain either more or less time in the buffer before being forwarded. In addition, when the buffer is small, the forwarding mechanism drops the oldest messages which are then excluded from the latency computation. Therefore, the delivery delay fluctuates as buffer size varies from 1000 kbytes and 5000 kbytes. After 5000 kbytes the latency has a tendency to remain constant. Interestingly, there is a point at which buffer size variation does not give any better performance. Note that with increasing node storage, competition for local storage is supposed to decrease so that SR is not invoked as often. As a result the overhead introduced by the SR migration process (see Section 3) is not significant and so the delivery delay decreases. In the case of AFNER and CCC the message flow increases and therefore message delivery delay decreases. In addition, AFNER and CCC can get more accuracy in their congestion control parameters due to the increase in message forwarding. Observe that, in the case of RRCC, by limiting replications the congestion control increases the message delivery delay. But as we increase the buffer size, messages that otherwise would not be delivered are.

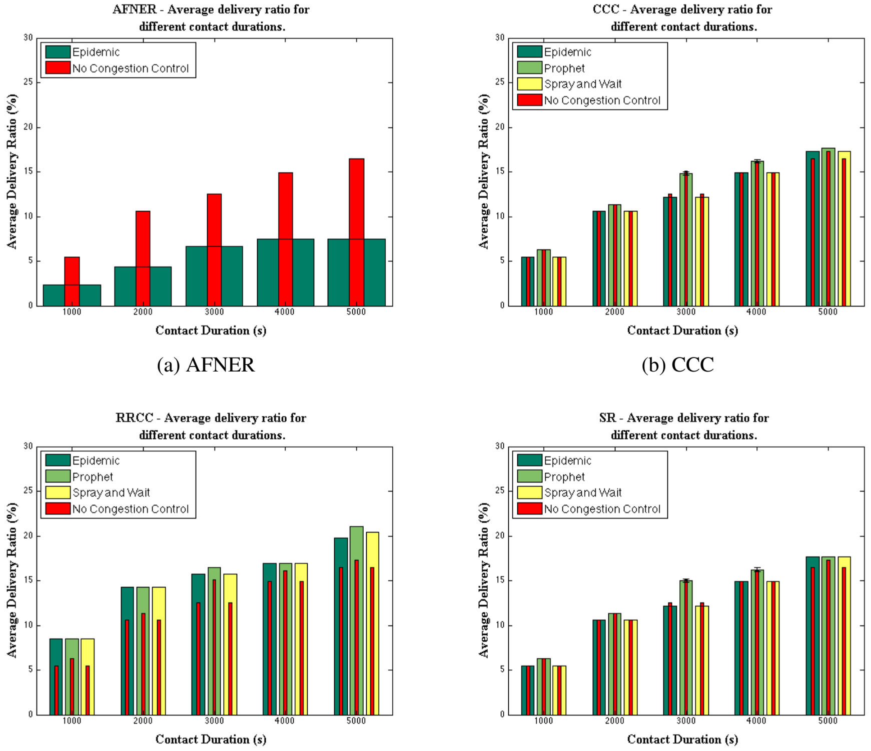
5.2. Different Contact Durations

In the next set of experiments, we use the same configuration parameters shown in Table 3. However, we introduce here a new parameter named contact duration. We fix the inter-contact time and vary the duration of the contacts. Table 6 shows an example of these configurations for one pair of nodes, namely *Base Station* and *Satellite 0*, where column *Duration* shows the contact duration which is managed by the *up* and *down* connection attributes and their times (columns *State* and *Time(s)*, respectively).

Figure 7 confirms that increasing the contact duration, thus making it possible to transfer more messages, increases the delivery ratio for all congestion control schemes. It is interesting to observe that RRCC yields higher delivery ratios when compared to AFNER, CC, and SR. We contend that RRCC's proactive AIMD algorithm contributes to RRCC's improved performance because it takes action before congestion becomes more severe. This reduces the number of message copies per node. These results also

Table 6: Contact duration for the encounters between *Base Station* and *Satellite 0*.

Duration	State	Time(s)	Initial Node	End Node
1000	up	2000	Base Station	Satellite 0
	down	3000	Base Station	Satellite 0
2000	up	2000	Base Station	Satellite 0
	down	4000	Base Station	Satellite 0
3000	up	2000	Base Station	Satellite 0
	down	5000	Base Station	Satellite 0
4000	up	2000	Base Station	Satellite 0
	down	6000	Base Station	Satellite 0
5000	up	2000	Base Station	Satellite 0
	down	7000	Base Station	Satellite 0



(a) AFNER

(b) CCC

(c) RRCC

(d) SR

Figure 7: Average delivery ratio for different contact durations (IPN scenario, transmit speed of 2.5 Mbps, buffer size of 4000 kbytes, contact duration 1000 s, inter-contact time 1000 s, and message generation period of 300 s; slim bars represent the network without any congestion control mechanism).

show the routing protocol independence feature observed in previous experiments.

6. Simulation Results for Terrestrial Scenario

This section presents the experimental results obtained from the terrestrial scenario. Specifically, the experimental scenario was considered with the selected congestion control mechanisms and is different from the IPN scenario because the terrestrial scenario uses different mobility models and a large number of nodes.

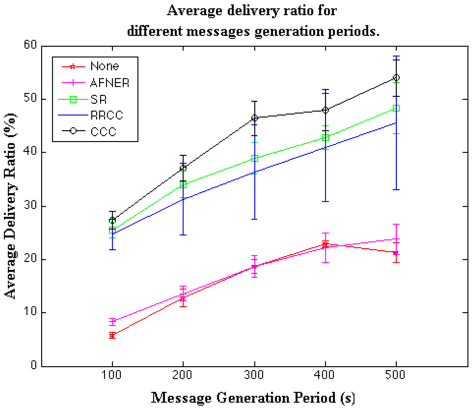


Figure 8: Message delivery ratio for terrestrial scenario as a function of the message generation period (Epidemic routing, buffer size of 1000 kbytes, transmit speed of 2.5 Mbps, and RWP mobility model).

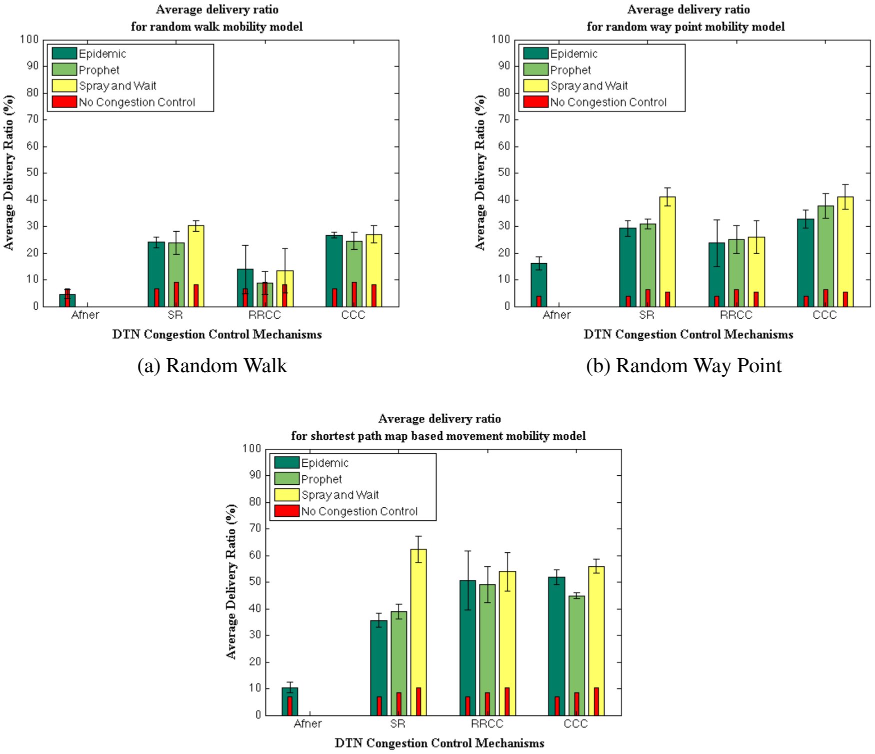
Figure 8 shows message delivery ratio as a function of the message generation period. As expected, for longer message generation periods, i.e., lower message generation frequencies, the delivery ratio increases. As expected, the results have similar trends as we have seen for IPN scenario. When comparing the results obtained from the IPN scenario (see Figure 3) with the terrestrial scenario, we clearly note that delivery ratios are superior for the terrestrial scenario since the selected mechanisms were designed for the terrestrial environment, as were the routing protocols. It is important to note here that the opportunism[2] of the routing protocols works better with a large number of nodes (we use 50 nodes in the terrestrial scenario and 5 nodes in the IPN scenario), enabling more opportunities for forwarding to arise. Observe that the CCC mechanism exhibits a higher average delivery ratio than the other mechanisms

[2] The word *opportunism* refers here to the routing protocol action guided primarily by self-interested motives. For instance, the routing protocol assumes the network has a large number of nodes resulting in a large number of contact opportunities and thus better opportunities to forward each message to its destination.

(AFNER, RRCC, SR). We believe CCC's higher delivery ratio is due to its hybrid (reactive and proactive) control. The proactive mode is prospective or future-oriented, helping to prepare the network for upcoming congestion through the predictive use of context (the refilling and refunding technique of CCC, see Section 3). In contrast, reactive control is retrospective, responding to the presence of salient or imperative congestion by engaging control only if needed, via reactivation of previously stored information (messages' credit). We believe that the good performance of CCC is due to the effective use of its credit-based strategy, that is, the CCC mechanism adopts a message self-contained congestion control. Furthermore, its refilling and refunding technique benefits from the large number of contact opportunities that arises in the experimental scenario. Note that AFNER has the worst performance (the smallest average delivery ratio) even when compared to the baseline scenario where it has the average delivery ratio values similar or equal to the values in the baseline scenario (see Figure 8 *None* curve). This happens because AFNER requires information about all nodes in the network. Getting this information poses certain challenges. For instance, the network connectivity changes dynamically which causes the network to be intermittently partitioned. Ideally we would like to have global network information to do congestion control, but this is not only inefficient and costly but usually unfeasible in DTN environments.

Node mobility is a critical factor influencing the performance of mobile networks. DTNs are no exception: the way nodes move determines their connectivity and thus impact their ability to relay messages.

Figure 9 shows the average delivery ratio for different mobility models and routing protocols, for each of the selected congestion control mechanism. As shown in Figure 9, we can see the influence of different mobility models on the average delivery ratio. In particular, when RW and RWP are considered, the delivery ratio is small. When we change the mobility model to SPMBM, the average delivery ratio for different protocols has a trend to increase. It is expected that the scenario with SPMBM (a denser network) has more contact opportunities. Thus, more messages can be forwarded, increasing the delivery ratio. However, the difference is minimal when we compare with RW and RWP. We argue that the delivery ratio becomes lower because



(a) Random Walk

(b) Random Way Point

(c) Shortest Path Map Based Movement

Figure 9: Average delivery ratio per congestion control mechanism for different mobility models (terrestrial scenario, transmit speed of 2.5 Mbps, buffer size of 500 kbytes, and message generation period of 300 s; slim bars represent the network without any congestion control mechanism).

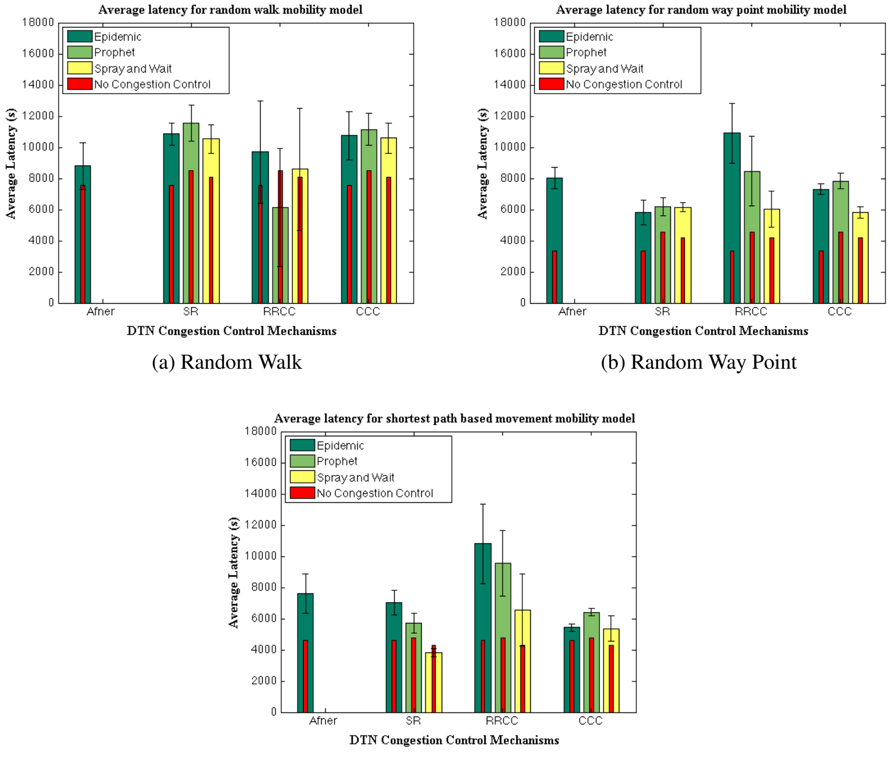
the buffers in the intermediate nodes on the shortest path may already be at maximum capacity. As a result, messages are discarded and the delivery ratio decreases. Furthermore, observe that AFNER performs worse than other mechanisms when compared to the baseline scenario. AFNER uses the network's average forwarding number to control congestion. Therefore, for networks that are more sparse (as is the effect of the RW and RWP mobility models) there is a large probability of disconnection (network partition). For this reason, more messages are waiting to be forwarded. As a result, buffers fill up and AFNER starts to discard messages based on an inaccurate network average forwarding number, which leads to a low delivery ratio. Unsurprisingly, this behavior confirms that the use of network global information to mitigate congestion in

DTN is not a good design principle. Furthermore, as previously pointed out, AFNER does not consider the number of message copies, which is one of the reasons for buffer overflow. Additionally, its reactive congestion control approach is not adequate to DTN environments.

As depicted in Figure 9, for all mechanisms except AFNER (which was designed to operate over epidemic routing protocol), when different routing protocols are used the average delivery ratio values vary. Therefore, in the view of the taxonomy presented in [9], RRCC, CCC and SR can be classified as routing-protocol-dependent (since they register different delivery ratio values for different routing protocols). It is noticeable that the CCC mechanism performs better with RW and RWP mobility models. When SPMBM model is used, CCC and RRCC exhibit similar behavior. We argue that their control strategies benefit from a denser network where there are more contact opportunities.

As expected, the congestion control mechanisms studied deliver better performance in the terrestrial appplications when compared to IPN scenarios which also confirms the hypothesis that no current congestion control approach is able to operate with similar performance in both scenarios.

In Figure 9b, we investigate the performance of the selected mechanisms when the RWP mobility model is used. All congestion control mechanisms achieved higher delivery ratios when compared with the baseline scenario. Looking at Figures 9a and 9b, we confirm that the mobility regime impacts congestion control performance. In particular, the mechanisms register higher delivery ratios for RWP. We attribute this behavior to the fact that, in the RWP model, mobile nodes are more likely to cluster in the geographical center of the simulation. Therefore , there are more contact opportunities and thus more messages can be forwarded. This increases the delivery ratio. At the same time, in the RWP model, node mobility and frequent but shorter contacts lead to higher buffer fill ratio, and thus more queuing. Overall, the gain in delivery ratio may or may not come with reduced delay. Specifically, SR and CCC mechanisms have better performance when comparing with other mechanisms, and note that they do not present good performance in the IPN scenario. The speed distributions in the RWP lead to a situation where at the stationary state each node stops moving. In this case, the movement executed by the nodes is similar to the scheduled movement in the IPN scenario. We argue that this leads to the good performance of the RRCC mechanism in both scenarios in a particular case. Therefore, in view of the above comparison, it may be feasible to have different congestion control mechanisms for each scenario.



(a) Random Walk

(b) Random Way Point

(c) Shortest Path Map Based Movement

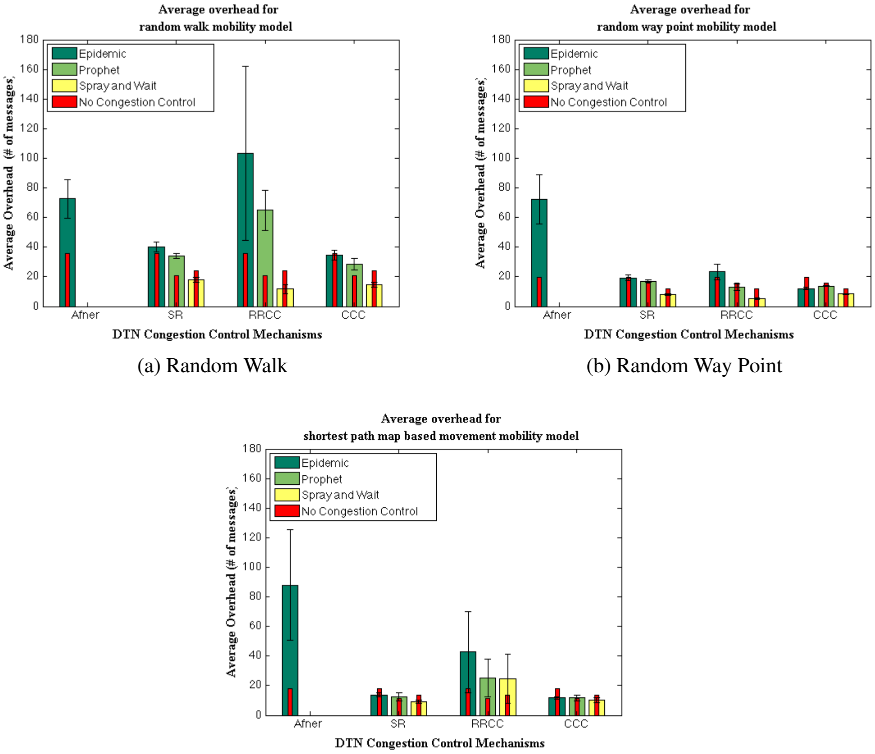
Figure 10: Average latency per congestion control mechanism for different mobility models (terrestrial scenario, transmit speed of 2.5 Mbps, buffer size of 500 kbytes, and message generation period of 300 s; slim bars represent the network without any congestion control mechanism).

Despite the network's higher density resulting from the SPMBM mobility model, we observe large average message delivery latency values as shown in Figure 10c. This is for the reason that messages may be forced to take longer routes (leading to larger delays) because well-connected nodes, which would enable shorter paths, may already have full buffers and thus cannot be chosen as next hops. Observe that mobile nodes choose random directions when using RW (Figure 10a) or RWP (Figure 10b).

In these trials, messages may travel over longer paths before arriving at their destinations, increasing the overall message average latency. The average latency decreases for SPMBM. The SPMBM model is an improved version of the RWP model, where nodes choose random destination decision points and move to those destinations following the map-based shortest path. Thus mobile nodes move less (due to the shortest path computation) when using SPMBM mobility model and, again, this model provides a denser network scenario in comparison to the RW and RWP, which are relatively sparse. As a result, in the SPMBM model the contacts are more frequent. Thus more messages are forwarded, reducing delivery delay and increasing delivery ratio. In all three mobility models, the combination of epidemic routing protocol and the AFNER mechanism results in relatively low average latency. AFNER's operation of dropping messages leads to reduced opportunity to forward older messages. The messages which have been discarded are those that have more copies in the network. As a result, newly arriving messages can be forwarded to their destination more rapidly, minimizing delivery latency. Again, though, AFNER uses the network's average forwarding number to decide which message to discard. This information can often be outdated because of intermittent connectivity, resulting in a low delivery ratio.

Figure 11 shows *overhead* incurred by each congestion control scheme under different routing regimes. It provides a measure of the average number of message transmissions required to deliver a message from the source to its destination. As such, the overhead graphic indicates the amount of network resources needed to deliver a message to its destination.

We expect that before congestion occurs, additional replicas will increase the probability that messages get delivered. However, as the network gets congested that relationship changes as more replicas mean more congestion and thus lower message delivery probability. As we have seen, Epidemic Routing results in the transmission of many more message copies within the network, compared to the Prophet and Spray and Wait protocols. This is because, under Epidemic Routing, each node replicates a message every time it encounters another node that does not have a copy of the message. This drastically increases the number of times a message is relayed to intermediate nodes. We also observe that AFNER's overhead is higher than the overhead incurred



(a) Random Walk

(b) Random Way Point

(c) Shortest Path Map Based Movement

Figure 11: Average overhead per congestion control mechanism for different mobility models (terrestrial scenario, transmit speed of 2.5 Mbps, buffer size of 500 kbytes, and message generation period of 300 s; slim bars represent the network without any congestion control mechanism).

by the other mechanisms. This behavior can be attributed to AFNER's use of Epidemic Routing as well as the fact that AFNER reactively discards messages based on the network's average forwarding number when congestion is detected. As a result, messages can be replicated and then until congestion has been mitigated. This increases the number of relayed messages in the network considerably while the number of delivered messages remains the same which increases the overhead. In other words, it increases the amount of network resources that are used to deliver one message to its destination. Note that for different mobility models the lower overhead values of the congestion control mechanisms yield higher delivery ratio values (see Figure 9).

7. Discussion

In order to evaluate the selected DTN congestion control mechanisms we implemented them as closely as possible to their original specifications. However, the results achieved varying levels of success. AFNER was originally implemented in the *ns-2* network simulator [13] using 50 nodes that move randomly at a speed of $20\ \mathrm{m/s}$. In the results reported, AFNER registered similar or worse performance compared with the no-congestion control baseline scenario in both the IPN and terrestrial applications. The main reasons for this behavior are: (1) AFNER's pure reactive approach to congestion control; (2) AFNER's use of epidemic routing; and (3) the fact that AFNER uses global information which is difficult to estimate accurately in DTNs.

In its original specification, RRCC [26]'s delivery ratio was evaluated as a function of buffer size and message generation period. Unsurprisingly, the delivery ratio increased for longer message generation periods. We observed the same behavior. In our study, RRCC's delivery ratio was higher in the terrestrial scenario and lower in the IPN scenario, which confirms that RRCC's design does not account for IPN's characteristics and operating conditions. Nonetheless, our results show that RRCC either outperforms or performs comparably to the other congestion control mechanisms we evaluated. We contend that RRCC's dynamic adjustment of the message replication degree as a function of perceived congestion is am effective way to combat congestion proactively as well as reactively. However, RRCC nodes adjust their replication degree based on the node's estimate of the global network's congestion state. Accurately estimating global state based on local information can be quite challenging. Additionally, RRCC's performance may suffer in scenarios where congestion is not caused by large number of message copies in the network but because of large message sizes.

The SR [23] mechanism, in its original specification, was evaluated as a function of buffer size since SR's congestion control depends on available buffer space at neighboring nodes. In DTN scenarios with long inter-contact times, which is the case of IPNs, finding neighbors may prove to be quite challenging. Furthermore, limited local storage and the fact that messages may remain stored at nodes for long periods of time poses additional challenges. Additionally, SR reacts to congestion by discarding messages when the local buffer is full and no available neighbors can be found, which can adversely affect its performance.

CCC [18] also had its performance evaluated in terms of delivery ratio as a function of buffer capacity. Recall that CCC is based on aging messages based on how many times they were forwarded and discarding older messages first. In CCC, messages are aged using refilling and refunding *credits* during node encounters. Our results confirm that CCC's performance is negatively impacted in sparse networks where node encounters are not frequent enough.

We should also mention the Contact Graph Routing (CGR) [5] initiative to address congestion control in DTNs with scheduled contacts. CGR takes advantage of a-priori knowledge of network topology and contact schedules. It has been implemented in the Interplanetary Overlay Network (ION) [4], an implementation of the DTN architecture. ION accomplishes congestion control in a "quasi-manual" fashion by computing *congestion forecasts* based on published contact plans. These forecasts are presented to the mission teams so that they can take corrective action, revising contact plans before the forecast congestion occurs. The transmission rates in the contact plans are enforced automatically by built-in rate control mechanisms in the ION sender, also known as ION bundle protocol agent. In the case of rate control failure, causing reception rate to exceed what was asserted in the contact plan, the receiving bundle protocol agent drops data according to a drop-tail policy to avoid congestion. The insertion of new bundles into the network can also lead to congestion. To avoid this, ION implements an admission control mechanism that may either function in a drop-tail manner or simply block the application until insertion of the new bundle no longer threatens to congest the node. We should note that while CGR's local congestion avoidance provides short-term solution for small DTNs, the general congestion problem in predictable DTN remains an open research topic. Furthermore, ION congestion control does not work in opportunistic DTNs.

8. Conclusion

This paper evaluated the performance of four DTN congestion control mechanisms which represent the state-of-the-art on DTN congestion control based on the survey presented in [9]. We examined performance in terms of delivery ratio, latency, and overhead. We also evaluated the congestion control schemes in terms of their routing protocol independence. As baseline, we use the same simulation scenarios without any congestion control. Although the selected congestion control mechanisms were not originally proposed for IPN scenarios, we argue that this kind of quantitative study provides useful insight to help guide the design of effective and interoperable DTN congestion control mechanisms.

Our study shows that, since messages may be buffered for long periods of time before being acknowledged, buffer overflow becomes highly common; this results in excessive latencies and message losses which is aggravated by the fact that reactive congestion schemes such as AFNER and SR would be severely impacted by delayed control decisions. Our results show that adopting a proactive or hybrid approach as done by RRCC and CCC may significantly improve performance; this appears to be an interesting strategy for future DTN congestion control mechanisms.

One interesting conclusion we must point out is that, due to short contact duration and longer inter-contact times, message expiration itself provides a degree of congestion control in the IPN scenario. In IPN operations, messages may expire while propagating to their destination or remain a long time in buffers waiting to be forwarded, so early discarding of those message may reduce buffer occupancy.

Our results also indicate that DTN routing protocols have significant impact on congestion control mechanisms' performance. Patterns of mobility show similar impact. The relation between routing and mobility appears clear here. Therefore, the design of more intelligent, efficient, and routing-protocol-independent congestion control mechanisms is of particular interest and should be considered.

In summary, our results provide important insight into designing new DTN congestion control mechanisms. They confirm that congestion control helps to increase message delivery ratio, even in highly congested network scenarios. Our study also indicates that good design principles for congestion control in DTN scenarios include: using a combination of reactive and proactive control as well as using local information instead of relying on global knowledge. Additionally, designing a congestion control mechanism that is routing protocol-independent helps with interoperability and applicability to a wide variety of DTN scenarios. A key challenge in DTN congestion control is to create a scheme that can provide good delivery performance and low delivery delay in DTNs that exhibit opportunistic-, scheduled intermittent contacts, or both.

8.0.0.1 Acknowledgements

Part of the research discussed in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

The Brazilian National Council for Scientific and Technological Development – CNPq (Process number 245492/2012-7) supported this project. And Coordenação de Aperfeiçoamento de Pessoal de nível Superior – CAPES (Process number BEX 5063/14-0) has also supported this project.

The authors would like to thank Dr. Shawfeng Dong from the Astronomy & Astrophysics Department at University of California Santa Cruz for hosting part of this research at the high performance computers of his department.

This work was partially funded by NSF under project CNS 1321151.

References

[1] Delay-tolerant networking research group. Available at: `https://sites.google.com/site/dtnresgroup/home` Accessed on: March 3rd, 2014.

[2] I.F. Akyildiz, O.B. Akan, Chao Chen, and Weilian Su Jian Fang. The state of the art in interplanetary internet. *IEEE Communications Magazine*, 42(7):108–118, July 2004.

[3] Giuseppe Araniti, Igor Bisio, and Mauro De Sanctis. Interplanetary networks: Architectural analysis technical challenges and solutions overview. In *IEEE ICC 2010 Proceedings*, pages 1–5, May 2010.

[4] Scott Burleigh. Interplanetary overlay network: An implementation of the DTN bundle protocol, 2007. JPL - California Institute of Technology, Pasadena, California.

[5] Scott Burleigh. Contact graph routing. Technical report, IETF-draft, 2012.

[6] Scott Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss. An approach to interplanetary internet. *IEEE Communication Magazine*, July 2003.

[7] Tracy Camp, Jeff Boleng, and Vanessa Davies. A survey of mobility models for ad hoc network research. *Wireless Communications and Mobile Computing*, 2(5): 483–502, 2002.

[8] C. Chen and Z. Chen. Towards a routing framework in ad hoc space networks. *International Journal of Ad Hoc and Ubiquitous Computing*, 5(1), 2010.

[9] Aloizio Pereira da Silva, Scott Burleigh, Celso Masaki Hirata, and Katia Obraczka. A survey on congestion control for delay and disruption tolerant networks. *Elsevier Ad Hoc Networks*, 2014. doi: 110.1016/j.adhoc.2014.07.032.

[10] Robert C. Durst, Gregory J. Miller, and Eric J. Travis. TCP extensions for space communication. In *Proceedings of the 2nd annual international conference on Mobile computing and networking*, pages 15–26, 1996.

[11] K. Holger and A. Willig. *Protocols and Architectures for Wireless Sensor Networks*. Wiley, 2005.

[12] Gerald James Holton and Stephen G. Brush. *Physics, The Human Adventure: From Copernicus to Einstein and Beyond*. Rutgers University Press, 2001.

[13] Information Science Institute. The network simulator (ns2). ISI: University of Southern California. Available at: `http://www.isi.edu/nsnam/ns/`, 2013. Accessed on March 23th, 2013.

[14] S. Jain, K. Fall, and R. Patra. Routing in delay tolerant network. In *Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Communications*, 2004.

[15] Vidya K and Hemanth S. R. Routing and congestion control strategies in opportunistic networks: A survey. *International Journal For Technological Research in Engineering*, 1(11), July 2014.

[16] Ari Keränen and Jorg Ott. Increasing reality for dtn protocol simulation. Technical report, Helsinki University of technology, Networking Laboratory, July 2007.

[17] Ari Keränen, Jörg Ott, and Teemu Karkkainen. The ONE simulator for DTN protocol evaluation. In *The 2nd International Conference on Simulation Tools and Techniques*, 2009.

[18] Lertluck Leela-amornsin and Hiroshi Esaki. Heuristic congestion control for massage deletion in delay tolerant network. In *Smart Spaces and Next Generation Wired/Wireless Networking*, August 2010. Third Conference on Smart Spaces and 10th International Conference.

[19] Sotirios-Angelos Lenas, Stylianos Dimitriou, Fani Tsapeli, and Vassilis Tsaoussidis. Queue-management architecture for delay tolerant networking. In *Proceedings of the 9th IFIP TC 6 International Conference on Wired/Wireless Internet Communications*, WWIC'11, pages 470–482. Springer-Verlag, 2011. URL `http://dl.acm.org/citation.cfm?id=2023094.2023141`.

[20] A. Lindgren, A. Doria, and et. al. Probabilistic routing in intermittently connected networks. *Computer Communication Review*, 7:19–20, 2003.

[21] Haider Mehdi, Safeeullah Soomro, W.R. Khan, Abdul Ghafoor Memo, and Abdul Hafeez. Error-rate performance analysis of wireless sensor networks over fading channels. *Sindh University Research Journal*, 45:177–182, 2013.

[22] Ioannis Psaras, Lloyd Wood, and Rahim Tafazolli. Delay-/disruption-tolerant networking: State of the art and future challenges. Technical report, Center for Communication Systems Research, Department of Eletrical Engeneering , University of Surrey, 2009.

[23] Matthew Seligman, Kevin Fall, and Padma Mundur. Alternative custodians for congestion in delay tolerant networks. In *Proceedings of the 2006 SIGCOMM workshop on Challenged Networks - CHANTS'06*, 2006.

[24] Aloizio P. Silva, Scott Burleigh, Celso M. Hirata, and Katia Obraczka. Congestion control in disruption-tolerant networks: a comparative study for interplanetary networking applications. In *Proceedings of the 2014 MobiCom workshop on Challenged Networks - CHANTS'14*, 2014.

[25] T. Spyropoulos, K. Psounis, and C.S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *ACM SIGCOMM workshop on Delay-tolerant Networking*, page 259, 2005.

[26] Nathanael Thompson, Samuel C. Nelson, Mehedi Baknt, Tarek Abdelzaher, and Robin Kravets. Retiring replicants: Congestion control for intermittently - connected networks. In *Proceedings of the IEEE INFOCOM'2010*, pages 1–9, March 2010.

[27] Amin Vahdat and David Becker. Epidemic routing for partially-connected ad hoc networks. Technical report, 2000.

[28] Dimitris Vardalis, Sotiris Diamantopoulosand Vassilis Tsaoussidis Sotiris Angelos Lenas, Francesco Belli, Philip Beavi, David Patterson, and Kim Nergaard. Decentralized space-data dissemination for low-cost, dense satellite networks. *IEEE Transactions on Aerospace and Electronic Systems*, 2015.

[29] Artemios G. Voyiatzis. A survey od delay and disruption tolerant networking applications. *Internet Engineering*, 5(1), 2012.

[30] R. Wang, T. Taleb, A. Jamalipour, and B. Sun. Protocols for reliable data transport in space internet. *IEEE Communications Surveys & Tutorial*, 11(2), 2009.

[31] Li Yun, Cheng Xinjian, Liu Qilie, and You Xianohu. A novel congestion control strategy in delay tolerant networks. In *Second International Conference on Future Networks*, 2010.

[32] Long Zhang and Xianwei Zhou. Hop-by-hop dynamic congestion control with contact interruption probability for intermittently connected deep space information networks. *Wireless Personal Communication*, pages 399–424, 2013.