Available online at www.sciencedirect.com
ScienceDirect
Elsevier Ad Hoc Networks 00 (2014) 1–17
Elsevier Ad Hoc Networks

Special Issue on New Research Challenges in Mobile, Opportunistic and Delay-Tolerant Networks

A Survey on Congestion Control for Delay and Disruption Tolerant Networks

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

[a]*Instituto Tecnológico de Aeronáutica,*
Department of Electronic and Computer Engineering
aloizio@ita.br, hirata@ita.br
[b]*University of California Santa Cruz*
Computer Engineering Department
katia@soe.ucsc.edu
[c]*Jet Propulsion Laboratory*
California Institute of Technology
scott.c.burleigh@jpl.nasa.gov

Abstract

Delay and disruption tolerant networks (DTNs) may experience frequent and long-lived connectivity disruptions. Unlike traditional networks, such as the TCP/IP-based Internet, DTNs are often subject to high latency caused by very long propagation delays (e.g., interplanetary communication) and/or intermittent connectivity. Another feature that sets DTNs apart from conventional networks is that there is no guarantee of end-to-end connectivity between source and destination. Such distinct features pose a number of technical challenges in designing core network functions such as routing and congestion control. In this paper, we survey the state-of-the-art in DTN congestion control. We propose a taxonomy to map the DTN congestion control design space and use it to classify existing DTN congestion control mechanisms.

Keywords: delay and disruption tolerant networks, interplanetary networks, congestion control, intermittent connectivity

1. Introduction

Delay and Disruption Tolerant Networks (DTNs) were initially motivated by the idea of deploying an Interplanetary Internet (IPN) [1] for deep space communication. As a result, a framework for an IPN which aims to use an interplanetary backbone to connect internetworks in space was developed. Over time, a diverse set of other DTN applications for "extreme" environments on Earth have emerged including vehicular networks, emergency response and military operations, surveillance, tracking and monitoring applications, and bridging the digital divide. In these applications, long delays are a consequence of the long distances and/or episodic connectivity which are characteristic of "extreme" environments.

The arbitrarily long delays and frequent connectivity disruptions that set DTNs apart from traditional networks imply that there is no guarantee that an end-to-end path between a given pair of nodes exists at a given point in time. Instead, nodes may connect and disconnect from the network over time due to a variety of factors such as mobility,

wireless channel impairments, nodes being turned off or running out of power, etc. Consequently, in DTNs, the set of links connecting DTN nodes, also known as "contacts", varies over time. This fundamental difference between DTNs and conventional networks results in a major paradigm shift in the design of core networking functions such as routing, forwarding, congestion and flow control.

The DTN architecture described in [2] uses the so-called *store-carry-and-forward* paradigm, as opposed to the Internet's *store-and-forward*, to deliver messages from source to destination. In *store-carry-and-forward*, nodes store incoming messages and forward them when transmission opportunities arise. Note that in traditional networks, nodes also store messages before forwarding them; however, the time scales at which data is stored locally while waiting to be forwarded are typically orders of magnitude smaller when compared to DTNs. Therefore, storage in *store-carry-and-forward* typically uses persistent storage which implies that DTN nodes need to be equipped accordingly.

According to the *store-carry-and-forward* paradigm, when a DTN node "encounters" another DTN node, it decides whether to forward messages it is carrying to the other node. Therefore, the concept of *links* in traditional networks (wired or wireless) is replaced with the notion of *contacts*. In scenarios where these encounters are random, *store-carry-and-forward* is also referred to as *opportunistic forwarding*. On the other hand, when contacts are known a priori (e.g., in deep space communication applications), *store-carry-and-forward* is known as *scheduled forwarding*. Finally, there are scenarios where node encounters follow a probability distribution based on past history; in these cases, *store-carry-and-forward* is based on *probabilistic forwarding* [3]. Note that, since *contact times* are finite and may be arbitrarily short, a node may need to choose which messages to forward based on some priority; a node may also decide whether the new neighbor is a "good" candidate to carry its messages. A node's "fitness" as a relay for a particular message depends on several factors that can be dependent on the message's ultimate destination (e.g., how often that potential relay encounters the destination, etc.); there are also factors that are destination-independent, for example, the relay's mobility patterns, its capabilities (e.g., storage, energy, etc.) [4] [5].

The simplest DTN forwarding technique is called *epidemic forwarding* [3] [6] [7] [8] [9], which is to DTNs what *flooding* is to traditional networks. To address issues such as limited contact times and limited network and node resources, several variants of "pure" *epidemic forwarding* [7] [10] [11] have been proposed. For instance, before a node forwards its messages to another node upon contact, the two nodes perform an initial "handshake" in which they exchange a summary of the messages each one has; then they only exchange messages that the other does not already carry. There are also a number of "controlled" epidemic variants that try to, implicitly or explicitly, limit the number of copies of the same message in the network.

The fact that in DTNs the existence of an end-to-end path between any pair of nodes at all times cannot be guaranteed raises fundamental challenges in end-to-end reliable data delivery. In DTNs, the Internet model of end-to-end reliability (as implemented by TCP) is not applicable. The DTN architecture proposed in [2] replaces TCP's end-to-end reliability with *custody transfer*, which uses hop-by-hop acknowledgements to confirm the correct receipt of messages between two directly connected nodes. Additionally, due to the inability to guarantee end-to-end connectivity at all times, functions based on the TCP/IP model such as congestion and flow control will not always work in DTNs. Instead, hop-by-hop control can be employed.

In this paper, we survey the state-of-the-art on DTN congestion control mechanisms. To this end, we propose a taxonomy to help (1) map the DTN congestion control design space and (2) compare existing DTN congestion control mechanisms. The survey presented in [12] considers reliability and congestion control proposals focusing on opportunistic networks. Note that opportunistic networks are a special case of DTNs where contacts between nodes are not known a priori. In our survey, we consider DTNs as more broadly defined: in addition to opportunistic networks, i.e., networks where contacts are random, we also explore networks in which contacts are scheduled well as networks in which contacts are probabilistic (based on some probability function derived from past contacts). The tutorial presented in [13] discusses the prospects of using DTN in future satellite networks, in particular LEO/GEO satellite constellations. Studies such as [14] and [15] confirm that congestion control is a fundamental issue in DTNs and note that it has not received much attention from the DTN research community. Our work goes a step further and provides a deeper analysis of existing DTN congestion control mechanisms.

The remainder of this paper is organized as follows. Section 2 provides an overview of the DTN architecture and discusses DTN congestion control comparing it against traditional Internet congestion control. In Section 3, we present a taxonomy for DTN congestion control and in Section 4, we describe existing DTN congestion control mechanisms in light of the proposed taxonomy. Section 5 provides design recommendations for future DTN congestion control mechanisms based on insights gained from our discussion of the current DTN congestion control state-of-the

art. Finally, Section 6 concludes the paper.

2. Background

2.1. The DTN Architecture

The DTN architecture, which was originally proposed in [16], aims at providing implementations for reliable message delivery in intermittently-connected networks. It introduces the *store-carry-and-forwarding* paradigm under which messages may remain stored for relatively long periods of time in persistent storage at intermediate nodes while in transit from source to destination. The DTN architecture was designed to operate as an intermediate layer, called the *bundle layer*, between the application and the transport layers of the networks it interconnects (see Figure 1). It provides services such as in-network data storage and retransmission, interoperable naming, authenticated forwarding, and coarse-grained classes of service.

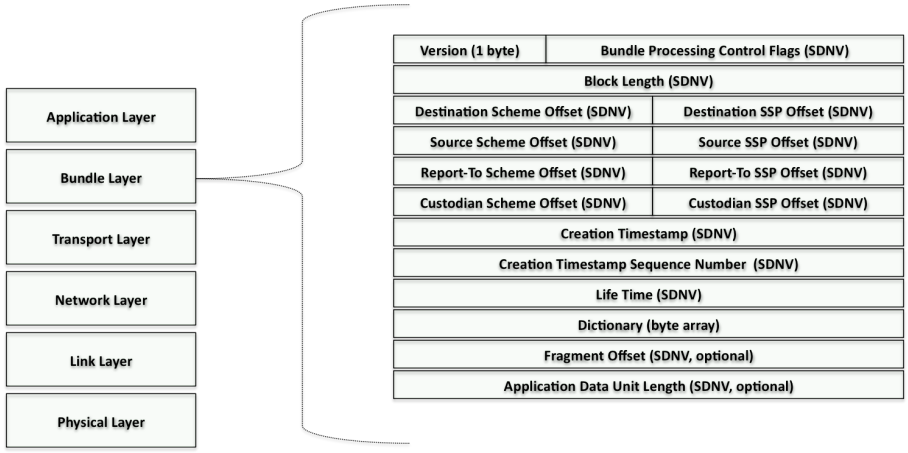

Figure 1. The DTN protocol stack and the structure of the primary block of a bundle.

The DTN architecture also specifies the *bundle protocol* [16, 17, 18, 19] which controls the exchange of *bundles*, i.e., application-layer messages. The Bundle Protocol can operate either atop transport protocols (e.g., TCP, UDP, etc), or atop lower layer protocols (e.g., Bluetooth, Ethernet, etc). The term "bundle" was chosen to connote the self-sufficiency of the messages: application-layer messages are expected to contain sufficient metadata to enable processing by the recipient without negotiation, as if all relevant metadata query and response messages have been anticipated by the sender and "bundled" into a single application data unit. When operating atop the transport layer, the bundle protocol receives messages from the application layer, encapsulates them into bundles, whose format is depicted in Figure 1, and then forwards them to the next-hop DTN node. The final destinations of bundles are "endpoints" associated with nodes and identified by EIDs (End Point Identifiers).

As previously pointed out, since end-to-end paths cannot be guaranteed, a DTN route consists of a series of time-dependent *contacts*, i.e., communication links that are established whenever nodes come in range of one another. Contacts may be parameterized by their duration, capacity, latency, end points, and direction. Also due to the inability to guarantee end-to-end routes, reliability is achieved hop-by-hop using *custody transfers*. A node taking custody of a message commits to deliver that message to its destination or another node that accepts the message's custody. Once the node reliably transfers the message to the message's next custodian, its responsibility as custodian for that message ceases. Custody transfers allow nodes to free up storage used to carry messages that get transfer to other nodes.

2.2. Internet Congestion Control

The Internet is a worldwide computer network that transmits data by packet switching based on the TCP/IP suite. The Internet architecture defines two transport-layer protocols for data transmission end-to-end, i.e., between two communicating processes running on two different hosts connected to the Internet. The first one is UDP (User

Datagram Protocol), which provides an unreliable data delivery service. In other words, the different pieces of the same application message may arrive out of order, appear duplicated, or go missing without notice. UDP does not provide congestion– or flow control capabilities. The other Internet transport protocol is TCP (Transmission Control Protocol). TCP provides reliable, ordered delivery of a stream of bytes between two communicating processes.

In addition to reliable delivery, TCP also performs flow and congestion control. At this point, we should make clear the difference between flow and congestion control. Flow control is "all about the receiver", i.e., it tries to ensure that the sender does not outpace the receiver, sending data faster than the receiver can receive. On the other hand, congestion control is "all about the network" making sure that the sender does not send more data than the network can handle.

Therefore, congestion occurs when resource demands from users/applications exceed the network's *available capacity*. The need for congestion control on the Internet surfaced in 1986 when the Advanced Research and Projects Agency Network (ARPANET), the precursor to the Internet, suffered congestion collapse [20]. Congestion collapse generally occurs at choke points in the network, where the total incoming traffic to a node exceeds the outgoing bandwidth. As described in [21], there are two fundamental approaches to the problem of controlling Internet congestion, namely: capacity provisioning and load control.

The **capacity provisioning** approach is based on ensuring that there is enough capacity in the network to meet the offered load. On the other hand, the **load control** approach ensures that the offered load does not exceed the capacity of the network. The latter approach inspired the development of the first Internet congestion control algorithm [22]. The basic idea behind the algorithm was to detect congestion in the network using *packet loss* as congestion indicator. Upon detecting a packet loss, the source reduces its transmission rate; otherwise, it increases it.

According to TRMG (Transport Modeling Research Group) [23] [24], performance metrics that can be used to evaluate Internet congestion control protocols include:

- **Convergence speed**: estimates time to reach the equilibrium, i.e., states how much time elapsed between the moment that the congestion was detected and the moment that congestion ceased to exist.
- **Smoothness**: is defined as the largest reduction in the sending rate in one RTT (Round-Trip Time). In addition, it reflects the magnitude of the oscillations through multiplicative reduction, which is the way TCP reduces its transmission rate.
- **Responsiveness**: is defined as the number of RTTs of sustained congestion required for the sender to halve the sending rate.
- **Fairness**: specify the fair allocation of resource between the flows in a shared bottleneck link.
- **Throughput**: characterizes the transmission rate of a link or flow typically in bits per second. Most congestion control mechanisms try to maximize throughput, subject to application demand and constraints imposed by the other metrics (network-based metric, flow-based metric and user-based metrics).
- **Delay**: can be defined as the queue delay over time or in terms of per-packet transfer times.
- **Packet loss rates**: measures the number of packets lost divided by total packets transmitted. Another related metric is the loss event rate, where a loss event consists of one or more lost packets in one round-trip time (RTT).

Some of the metrics discussed above could have different interpretations depending on whether they refer to the network, a flow, or a user. For instance, throughput can be measured as a network-based metric of aggregate link throughput, as a flow-based metric of per connection transfer times and as user-based utility metric.

These metrics were originally proposed for the Internet and in [25], they have been used to provide a categorized description of different congestion control strategies in packet networks using network-awareness level as a criterion. While some of them can still be used for DTNs, new metrics are needed. For instance, in DTNs, queueing delays are expected because of the high latencies and intermittent connectivity. Furthermore, paths are lossy, so losses do not necessarily indicate congestion as assumed in TCP. In Section 5, we discuss metrics employed by DTN congestion control protocols as well as metrics to measure their performance.

2.3. Congestion Control in DTN: Challenges

As previously pointed out, the challenges of controlling congestion in DTNs are mainly due to two reasons: (1) episodic connectivity, i.e., end-to-end connectivity between nodes cannot be guaranteed at all times, and (2) communication latencies can be arbitrarily long caused by high propagation delays and/or intermittent connectivity. Consequently, traditional congestion control does not apply to DTN environments. A notable example is TCP's congestion control mechanism which is not suitable to operate over a path characterized by extremely long propagation delays, particularly if the path contains intermittent links. Basically, TCP communication requires that the sender and the receiver negotiate an end-to-end connection that will regulate the flow of data based on the capacity of the receiver and the network. Establishment of a TCP connection typically takes at least one round-trip time (RTT) before any application data can flow. If transmission latency exceeds the duration of the communication opportunity, no data will be transmitted [16] [26]. Furthermore, there is a two-minute timeout implemented in most TCP stacks: if no data is sent or received for two minutes, the connection breaks. However, in some DTNs RTTs can be much longer than two minutes.

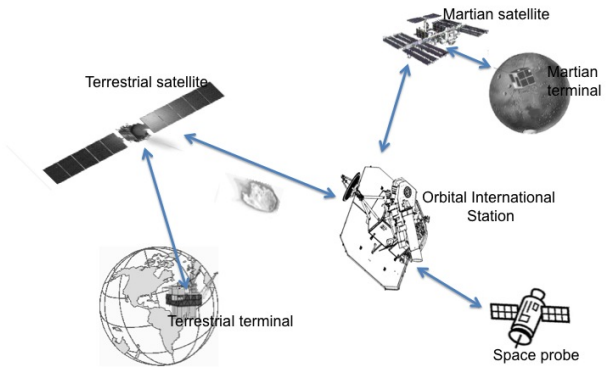

Figure 2. DTN connection example: deep space communication scenario

For example, in an interplanetary network supporting Earth-Mars communication (as illustrated in Figure 2), the RTT is around 8 minutes when both planets are closest to one another, with a worst-case RTT of approximately 40 minutes. In this scenario, the terrestrial satellite is connected to an Orbital International Station (OIS) in orbit around the Sun, which in turn connects to a Martian satellite and a space probe. Additionally, there is a Martian terminal connected to the Martian Satellite. The link between the OIS and the Martian Satellite is interrupted whenever the planet Mars is between the OIS and the orbiting satellite, as well as whenever the Sun is between Mars and the OIS. Therefore, traffic on the "link" between the Terrestrial and Martian satellites may need to be buffered at the OIS for long and varying periods of time. If the OIS becomes heavily congested, it will significantly hamper communication between the Terrestrial satellite and the space probe.

Alternatives such as UDP or DCCP (Datagram Congestion Control Protocol [27]) are not generally appropriate because they offer limited reliability and, in the case of UDP, no congestion control. Consequently, efficient techniques are thus needed to effectively control congestion in DTNs so that network utilization is maximized.

3. DTN Congestion Control Taxonomy

One of the contributions of this survey is to propose a taxonomy to classify DTN congestion control mechanisms. The proposed taxonomy (see Figure 3) maps the DTN congestion control design space and uses it as backdrop to put existing DTN congestion control techniques in perspective. We also discuss possible directions for future exploration of efficient congestion control for a range of DTN applications and scenarios. In this section, we present the criteria underpinning the proposed DTN congestion control taxonomy.

3.1. Congestion Detection

How is congestion detected? In general, congestion occurs when resource demands exceed available capacity. Consequently, congestion detection can consider: *network capacity*, *buffer availability*, and *drop rate*.

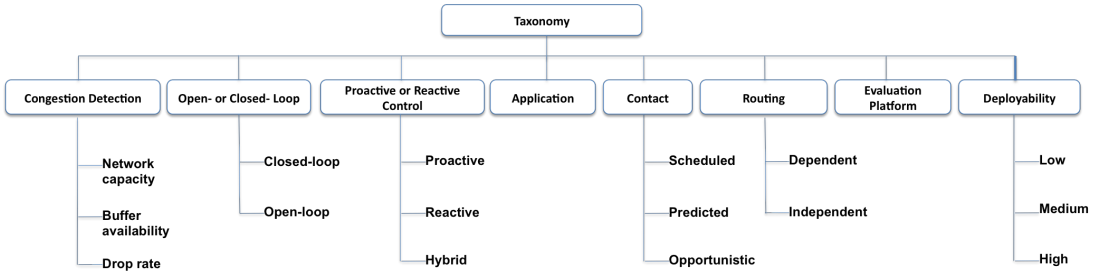

Figure 3. Overview of the proposed taxonomy

- *Network capacity*: mechanisms that use network capacity to gauge congestion try to assess if the traffic arriving at the network is greater than the traffic that could be supported by network.
- *Buffer availability*: some DTN congestion detection strategies check whether the storage capacity at nodes is filling up as new messages are received.
- *Drop rate*: the fact that the drop rate goes beyond a certain threshold is used by certain mechanisms as a way to detect congestion.

3.2. Open- versus Closed-Loop Control

Another way to classify DTN congestion control mechanisms is based on whether they employ *open loop*– or *closed loop* control. In open-loop congestion control, end systems do not rely on feedback from the network; instead they try to "negotiate" their sending rate a priori which can be considered a congestion prevention approach. Closed-loop (or *reactive* control) mechanisms utilize feedback from the network to the end systems. Network feedback usually contains information about current conditions. End systems typically respond to congestion build-up by reducing the traffic load they generate [28]. Notable examples of open-loop congestion control mechanisms include the ones based on buffer management, e.g., dropping messages according to a variety of policies. Some of these techniques use classical drop policies (i.e. drop-random and drop-tail) which were inherited from traditional networks. In section 4.2 below, we list and discuss some of these classical policies as well as drop policies which have been proposed specifically for DTNs (i.e., E-Drop and Mean-drop).

3.3. Proactive versus Reactive Control

Congestion control mechanisms can also be classified as *proactive* or *reactive*. Proactive congestion control (also known as congestion avoidance) schemes take a preventive approach and try to prevent congestion from happening in the first place; in reactive congestion control, end systems typically wait for congestion to manifest itself (e.g., router queue build-up, packet loss, etc.) before any action is taken.

Because DTN systems may exhibit long delays, reactive congestion control may not be sufficient. Proactive congestion control or hybrid approaches combining proactive– and reactive control are interesting alternatives.

We should also point out that, at first glance, proactive– and open-loop control may seem to subsume one another. However, a closer look reveals that approaches based on open-loop control can be reactive. For instance, we can have a mechanism that is triggered solely based on the size of the router's queue, and therefore is open-loop. However, it can still be reactive if it starts dropping packets only when the queue is full. In a proactive open-loop approach, packets would start getting dropped sooner in an attempt to avoid letting congestion settle in.

3.4. Application

DTNs have a wide range of applications from deep space communication to mobile sensor networks on earth. Depending on the application, DTNs may exhibit very different characteristics to be able to address the requirements of the driving applications. Therefore, in our congestion control taxonomy, we consider the DTN application as an important classification criterion.

3.5. Contacts

When a communication opportunity appears between two DTN nodes we call this a *contact*. There are different kinds of contacts [29] [30] depending on whether they can be predicted. *Scheduled* contacts are predictable and known in advance. A typical example of a DTN with scheduled contacts is interplanetary networks, in which the mobility of nodes is completely predictable and known a priori. In *probabilistic* DTNs, contacts are probabilistic following a particular distribution that is based on historical data. Finally, contacts in *opportunistic* DTNs are totally random and not known ahead of time.

3.6. Routing

In our taxonomy, we also consider whether there is any relationship between congestion control and routing. We then classify congestion control mechanisms as routing protocol- *independent* or *dependent*. Congestion control approaches that can work with any routing mechanism are said to be routing-protocol independent; on the other hand, congestion control mechanisms that are proposed taking into account specific DTN routing protocols are routing-protocol dependent.

An interesting aspect of protocol-dependent congestion control is that, often times, the same mechanism that is employed for routing also serves the congestion control mission. A simple example of such "serendipitous" congestion control is controlled epidemic, where the same mechanism used to limit the number of message copies in the network is also performing congestion control.

3.7. Evaluation Platform

Most experimental evaluations of proposed DTN congestion control mechanisms have employed network simulation platforms. A significant advantage of network simulators is the fact that they make it easy to subject protocols under evaluation to a wide range of network and traffic conditions. They also allow experiments to be reproduced easily. In our taxonomy, where applicable, we include information about the simulation platform used to evaluate a particular congestion control mechanism.

3.8. Deployability

In the context of DTN congestion control, we define *deployability* as the ability to deploy a protocol in realistic scenarios under real-world conditions. As previously pointed out, most existing DTN congestion control protocols have been implemented and tested using simulation platforms, which do not necessarily expose the mechanisms being evaluated to real-world conditions. In order to explore the deployability of existing congestion control mechanisms, in Section 4.8, we classify them using three different levels, namely: low-, medium, and high deployability. The criterion we use here to assess the deployability of DTN congestion control mechanisms is whether they relay on global knowledge of the network.

4. Classification of DTN Congestion Control Mechanisms

In this section, we provide an overview of existing DTN congestion control mechanisms in light of our DTN congestion control classification. We summarize the various mechanisms according to our taxonomy in Table 6.

4.1. Congestion Detection

Congestion detection techniques and the congestion indicators they use are critical to the congestion control effort. They determine how reliably a network is able to detect congestion and how quickly it is able to react to it.

Several DTN congestion control mechanisms use buffer occupancy rate, that is, the availability of buffer space, as an indicator of congestion [31][32] [33] [34] [35] [36] [37] [38] [39] [40] [41] [42] [43] [44] [45] [46] [47] [48] [49].

As an example, the congestion control technique proposed in [42] checks buffer availability of potential DTN custodians, i.e., nodes that will be assuming responsibility for carrying messages on behalf of other nodes. If the custodian's buffer is full or the new message will not fit, the DTN custodian is considered to be congested.

Network capacity has also been used as congestion indicator in DTNs [50] [51] [52]. Token based congestion control [50] tries to regulate the amount of traffic entering a network based on network capacity, where network capacity is measured by the amount of data a network can deliver to destinations in a given period of time.

Some efforts employ number of dropped packets à la TCP as a way to detect congestion [53] [54]. However, due to frequent topology variations and high error rates, packets may be dropped for reasons other than congestion. For this reason, the congestion detection proposed in [53], partially follows TCP's strategy: the proposed congestion indicator is a function of the volume of message drops and replication.

4.2. Open- versus Closed-Loop Control

As illustrated in Table 6, over 80% of the DTN congestion control mechanisms considered here [55] [43] [32] [38] [36] [37] [50] [42] [46] [34] [53] [39] [41] [31] [54] [35] [40] [48] [49] [52] can be classified as closed loop. For example, the approach described in [38] is based on the observation that congestion in DTN nodes builds up gradually. It then proposes three states for DTN nodes, namely: normal state, congestion adjacent state (CAS), and congestion state (CS). Congestion control is then performed depending on the state of the node. More specifically, the proposed congestion control mechanism adopts a closed loop approach by having nodes broadcasting congestion information to their neighbors once they enter into CAS or CS.

The other mechanisms we examined, i.e., [33] [47] [51] [45] [44], fall into the open-loop control category. For example, in [51] a congestion control strategy called the Average Forwarding Number based on Epidemic Routing (AFNER) is proposed. In AFNER, when a node needs to receive an incoming message and its buffer is full, the node randomly drops a message from those whose forwarding number is larger than the networks average forwarding number. The forwarding number of a message is defined as the number of copies that have been made of a message, and the average forwarding number is the mean forwarding number of all the messages currently in the network. According to the forwarding number queue, the strategy determines the packet forwarding sequence. Open-loop congestion control mechanisms are generally based on dropping messages, while closed-loop approaches use feedback information, i.e., messages from neighbors, to avoid having to drop messages. They typically only drop data as a last resort. Some existing DTN congestion control strategies use drop policies that have been proposed for "traditional" networks. A list of some traditional drop policies along with the DTN congestion control mechanisms that use them is showed in Table 1.

Table 1. Traditional drop policies

Drop Policy	Description
Drop-random [56][31][36][41]	a message from the queue is selected at random to be dropped.
Drop-head [56][43]	the first message in the queue, i.e., the head of the queue, is dropped.
Drop-tail [56][43][32] [42][50] [52]	the most recently received message, i.e., the tail of the queue, is removed.
NHop-Drop [51]	any message that has been forwarded over N hops is dropped.
Drop-least-Recently-received [56][46]	the message that has been in the node buffer longest is removed.
Drop-oldest [56][39][43]	the message that has been in the network longest is dropped.
Drop-youngest [56]	drops the message with the longest remaining life time.

Table 2 show some drop policies (and the mechanisms that use them) which have been proposed specifically for DTNs.

While one could argue that open loop congestion control systems are a better match for DTNs, most existing mechanisms employ a closed loop approach.

4.3. Proactive versus Reactive Control

Congestion control solutions can be classified as proactive (i.e., performs congestion avoidance), reactive (i.e., responds to congestion events), or hybrid (i.e., performs congestion avoidance and reacts to congestion build-up).

From Table 6, we observe that most existing DTN congestion control mechanisms (around 52%) adopt a hybrid policy using both proactive– and reactive control [31] [32] [33] [36] [37] [38] [40] [55] [53] [46] [51] [48] [49]. An example of a hybrid approach that combines random early detection (RED) and explicit congestion notification (ECN) within the DTN architecture is described in [31].

We also note that approximately 31% of mechanisms studied adopt a purely reactive approach [34] [35] [42] [43] [54] [44] [45], while 17% of them are based on a purely proactive policy [50] [39] [41] [47] [52]. The approach proposed in [39] proactively advertises buffer occupancy information to adjacent nodes. Nodes can then avoid forwarding messages to nodes with high buffer occupancy. Following the mechanism described in [42], when a DTN node becomes congested, it tries to migrate stored messages to alternate locations to avoid loss.

Table 2. DTN drop policies

Drop Policy	Description
Drop-largest [57]	the message of the largest size is selected.
Evict Most Forwarded First - (MOFO) [58]	the message forwarded the maximum number of times is selected.
Evict Most Favorably Forwarded First - (MOPR) [58][51]	each message is related to a forwarding predictability FP. Whenever the message is forwarded the FP value is updated and the message that contains the maximum FP is dropped first.
Evict Shortest Life Time First - (SHLI) [58][39]	the message that contains the smallest TTL is dropped.
Evict Least Probable First - (LEPR) [58]	since the node is less likely to delivery a message for which it has a low P-value (low delivery predictability) and that it has been forwarded at least MF times, drop the message for which the node has the lowest P value.
Global Knowledge Based Drop - (GBD) [59]	based on global knowledge about the state of each message in the network (number of replicas), drop the message with the smallest utility among the one just received and the buffered messages.
History Based Drop - (HBD) [59]	a deployed variant of GBD that uses the new utilities based on estimates of m (the number of nodes, excluding the source, that have seen message since its creation until elapsed time) or n (the number of copies of message in the network after elapsed time). The message with the smallest utility is selected.
Flood Based Drop - (FBD) [59]	accounts only for the global information collected using simple message flooding, that is, without considering message size.
Threshold Drop - (T-Drop) [60]	drops the message from congested buffer only if size of existing queued message(s) falls in Threshold range (T).
Equal Drop - (E-Drop) [61]	drops the stored message if its size is equal to or greater than that of the incoming message; otherwise does not drop.
Message Drop Control - (MDC) [62]	the largest size message will be dropped. This policy controls the message drop through the use of an upper bound.
Mean Drop [63]	drops messages that have size greater than or equal to the mean size of queued messages at the node.
Small-Copies Drop [64]	drops messages with the largest expected number of copies first.
Adaptive Optimal Buffer Management Policies - (AOBMP)[65]	drops messages according to the utility function associated to the message.

4.4. Application

The challenging peculiarities of interplanetary networking environments inspired the congestion control mechanisms proposed in [31] [35] [55] [52] which target deep space communication scenarios. In the case of [35], to evaluate the Linklider Transmission Protocol - Transport (LTP-T), a scenario that emulates a deep-space network consisting of nodes on Earth and around Mars was modeled. In this context, the proposed congestion control protocol is designed to withstand the noise and delays incurred by communication across astronomical distances.

Terrestrial applications are considered in [43] [32] [38] [36] [37] [50] [39] [41] [53] [48]. In [53] one of the scenarios discussed models the behavior of mobile agents in disaster relief operations.

Dynamic scenarios with random generation of nodes following a statistic model are used in [33] [34] [54] [40] [44] [42] [47] [51] [45] [46] [49]. Most of them are based on using mobile terrestrial communication applications to study the proposed congestion control mechanisms. In the case of [33] the authors use a simple scenario where nodes and networks parameters are generated randomly and a mobility model is set.

4.5. Contacts

We observe from Table 6 that around 17% of DTN congestion control mechanisms [53] [45] [43] [33] assume in their scenarios both predicted and opportunistic contacts. Among the remaining congestion control mechanisms [31] [35] [55] [52], another 13% assume scheduled contacts. In particular, the hop-by-hop based mechanism proposed in [55] experiments with an interplanetary network scenario which employs scheduled contact between planets.

Approximately 39% of the mechanisms we studied, namely [34] [39] [40] [44] [46] [51] [47] [48] [49] assume opportunistic contacts while around 30%, namely [32] [36] [37] [38] [41] [42] [50] [54] use predicted contacts.

One example of opportunistic contact can be seen in [34] where a congestion control routing algorithm for security and defense based on social psychology and game theory is presented. In this case, DTN nodes are assumed to be randomly distributed and to perform random routing. Random routing results in randomness of each node's encounters.

A distributed congestion control algorithm that adaptively chooses the next-hop based on contact history and statistics is described in [36]. It has a component called contact manager that executes a forwarding heuristic taking into account predicted contacts.

4.6. Routing

Most of the DTN congestion control mechanisms, e.g., [31] [35] [36] [37] [38] [41][43] [53] [54] [44] [45] [46] [51] [48] [49], were proposed taking into account specific DTN routing protocols. We have classified them as routing protocol dependent. For example, the AFNER approach described in [51] proposes a congestion control strategy for epidemic routing. On the other hand, there are mechanisms such as [32] [33] [34] [50] [39] [40] [55] [42][43] [53] [47] [52] that are independent of the routing protocol. This means that congestion control does not act- or rely on the underlying routing mechanism and thus can be used with any routing infrastructure. A notable example is the Credit-Based Congestion Control mechanism [40] which uses the age of a message as a heuristics to decide when the message will be discarded when congestion builds up in nodes.

We should point out that for DTNs that require interoperability through the Bundle Layer [18], congestion control mechanisms should be independent of the routing protocol. Intuitively, congestion control mechanisms that work independently from the underlying routing protocol are more general and applicable to a wide array of scenarios.

4.7. Evaluation Platform

Approximately 17% of the mechanisms covered in this survey were evaluated using custom simulators [32] [36, 37] [50] [47]. For example, a discrete event-driven simulator was developed in [47] to test a congestion management strategy that uses the concept of revenue management and employs dynamic programming.

Another 17% of the techniques surveyed were evaluated using the *ns-2* network simulator [66], a simulation platform widely used in network research [55] [39] [31] [51].

Almost 44% of the approaches we researched [34] [44] [45] [53] [41] [54] [46] [40] [48] [49] use the ONE simulator [67]. ONE was designed specifically to evaluate DTN protocols and has become popular within the DTN research community. For instance, a local approach to detect and respond to congestion by adjusting the copy limit for new messages [54] has been implemented and evaluated using the ONE simulator.

The remaining approaches we surveyed employ other simulator platforms such as OPNET [38], YACSIM [43], GT-ITM [43], Weka [33], Netem [35] and ION [52] [68].

We should also highlight the Interplanetary Overlay Network (ION) [68] implementation of the DTN architecture. ION accomplishes congestion control 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 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.

The general trend we observe is that most proposed DTN congestion control mechanisms have been evaluated experimentally using simulation platforms. Employing more realistic experimental environments including real world scenarios and testbeds should become a priority in DTN protocol research.

4.8. Deployability

DTN congestion control mechanisms that rely on global network knowledge are classified as “low” deployability. A notable example is the protocol proposed in [51] which uses the network’s average forwarding number, i.e., the average forwarding number considering all messages currently in the network, to decide which message(s) to drop in case nodes get congested (see Table 3). The low deployability of this approach is due to the fact that, in order to compute the average forwarding number, nodes need global knowledge of the network, which is hard to acquire, especially in DTNs.

Table 3. DTN congestion control mechanisms with low deployability

Mechanism	Description
Average forwarding number based on epidemic routing (AFNER) [51]	Nodes drop messages whose forwarding number is larger than the network’s average forwarding number (a message’s forwarding number is the number of copies of that message floating in the network).

Around 70% of the surveyed DTN congestion control mechanisms ([43] [38] [36] [37] [50] [42] [46] [34] [39] [41] [31] [35] [55] [53] [54] [45] [44] [49]) can be classified as "medium" deployability. For example, the approach presented in [54] tries to respond to congestion by adjusting the maximum number of message copies based on the current level of network congestion. Nodes estimate global congestion levels using the ratio between drops and duplicate deliveries obtained during node encounters. From Table 4, which shows a brief description of medium-deployability mechanisms, we observe that such mechanisms rely on local neighborhood information to perform congestion control.

Table 4. DTN congestion control mechanisms with medium deployability

Mechanism	Description
Hop-by-hop Local Flow Control [55]	Nodes use hop-by-hop flow control where the sender verifies if the link is active and if there is resource availability towards the next-hop receiver.
Storage Routing for DTN Congestion Control [43]	Under congestion, nodes use a migration algorithm to transfer messages to other, less congested nodes.
Congestion Avoidance Based on Path Avoidance [38]	This scheme manages node storage and defines three states: normal, congestion adjacent, and congested. Nodes broadcast their current state to their neighbors who avoid forwarding messages to congested nodes.
Context Aware Forwarding Algorithm (CAFÈ) [36] [37]	Nodes adaptively choose a message's next hop based on contact history and statistics.
Token Based Congestion Control [50]	In this scheme all nodes must have a token in order to inject messages into the network. Tokens are initially uniformly distributed and after some time move randomly throughout the network.
Push-Pull Custody Transfer [42]	This approach uses buffer space availability information from neighbors to mitigate congestion. It includes a set of algorithms to determine which messages should be migrated to which neighbors and when.
Incentive Multi-Path Routing with Alternative Storage (IMRASFC) [46]	This scheme uses an incentive mechanism to stimulate mal-behaving nodes to store and forward messages and also try to select alternative neighbors nodes with available storage.
Congestion Control Routing Algorithm for Security Defense based on Social Psychology and Game Theory (CRSG) [34]	This approach uses social psychology and game theory to balance network storage resource allocation. It does this by obtaining node buffer utilization during node encounters.
Node-Based Replication Management (RRCC) [53]	This scheme detects and responds to congestion by adjusting the message copy limit. It uses local measurements, e.g., the ratio of the number of dropped messages to the number of message replicas measured by a node.
Congestion Avoidance Based on Buffer Space Advertisement [39]	Nodes advertise their buffer occupancy to adjacent nodes; neighboring nodes then use this information to decide their next hop when forwarding messages.
Congestion Aware Forwarding [41]	Congestion avoidance strategy which utilizes heuristics to infer shortest paths to destinations from social information (e.g., connectivity); it uses buffer occupancy and communication latency information to avoid areas of the network that are congested.
Combined Congestion Control for IPN [31]	This mechanism combines ECN (Explicit Congestion Notification) and RED (Random Early Detection) with storage-based routing strategies and makes use of neighbor buffer occupancy to mitigate congestion.
Threshold-Based Congestion Control [54]	This scheme tries to respond to congestion by adjusting the copy limit for messages based on the current observed network congestion level. The congestion level is estimated based on information collected during node encounters.
Lincklider Transmission Protocol Transport (LTP-T) [35]	LTP-T maintains a congestion timer for forwarded blocks. The idea is that an entire block has to be transmitted before the timer expires. If the congestion timer expires at an intermediate node, this node should signal the presence of congestion to upstream nodes using LTP-T's congestion notification.
Simulated Annealing and Regional Movement (SARM) [45]	SARM adopts a message deleting strategy based on regional characteristics of node movement and message delivery. As nodes move, the algorithm records the cumulative number of encounters with other nodes. When congestion happens, the cumulative number of the node as the transferred node and the destination node for all messages in these two nodes is calculated and then the message is selected to be deleted.
Following Routing (FR) [44]	FR assumes all nodes are mobile; if a node A tries to relay a message to node B but B is not able to receive the message (e.g., because its buffer is full), A tries to follow B's trajectory hoping to encounter a suitable next hop or the destination node.
Dynamic Congestion Control Based Routing (DCCR) [48]	DCCR is based on replication quotas. Each message is associated with an initial quota. During its time-to-live, a message can have its quota value updated according to a quota allocation function. This function reduces or increases the message's replication quota. Moreover, each node maintains buffer occupancy and contact probability with other nodes. In this case each node can compute the local measurement of congestion level information in order to update replication quotas.

The remainder of the mechanisms we investigated, namely [32] [33] [40] [47] [48] [52] (see Table 5) can be in-

cluded in the high-deployability category since they try to perform congestion control by using only local information, i.e., information about the node itself. For example, the scheme presented in [47] proposes a congestion management technique that decides whether to accept custody of a bundle if and only if the benefit of accepting custody is greater than or equal to the cost of the resources used to store the bundle.

Table 5. DTN congestion control mechanisms with high deployability

Mechanism	Description
Autonomous Congestion Control [32]	This mechanism adopts a financial model and compares the receipt and forwarding of messages to risk investment. When a new message arrives, the node decides whether to receive it or not according to a risk value of receiving and storing the message. The risk value is determined by local metrics, such as the node's own buffer space and the input data rate.
Distributive Congestion Control for Different Types of Traffic [33]	Congestion control is accomplished by distributing traffic according to different priority levels. Messages with higher priority are ensured minimum bounded delay whereas those with lower priority are discarded at higher congestion levels.
Congestion Management Based on Revenue Management and Dynamic Programming [47]	Nodes accept custody of messages (or bundles) if and only if there is sufficient remaining resource capacitiy and the resulting benefit of acting as relays is greater than or equal to the cost associated to the use of local resources.
Credit-Based Congestion Control [40]	According to this strategy, some credit is associated to each message; when a two nodes encounter each other, the amount of credit for each message is updated. When buffers become full, messages that have the least credit are dropped.
Message Admission Control Based on Rate Estimation (MACRE) [49]	This congestion control scheme decides whether to admit a message according to the relationship between a node's input rate and output rate.
Interplanetary Overlay Network (ION) [68] [52]	ION's congestion control is anticipatory and is performed based on the "contact plan" between the nodes. The maximum projected occupancy of a node is based on the computation of the congestion forecast for the node. Thus, congestion control is performed essentially manually.

5. DTN Congestion Control Design Guidelines

The majority of the congestion control techniques we investigated were developed for, and validated in, specific scenarios. Their applicability is therefore limited in various ways. For example, some control techniques depend upon anomalous conditions that are difficult to reproduce in an operating DTN environment but that induce temporary congestion in very special situations. Consequently, these mechanisms may not accurately manage routine congestion in a nominally operating network.

As previously pointed out, most existing congestion control protocols have been evaluated experimentally using simulation platforms and some "traditional" performance metrics, i.e., metrics used to evaluate Internet congestion control as discussed in Section 2.2. DTN-specific performance metrics have been proposed including buffer occupancy and message replication. Additionally, DTN congestion control techniques have also employed "DTN-aware" indicators to help in the congestion control effort, e.g., social behavior of nodes, node mobility, message retention in storage, node encounter probability, node connectivity and betweenness, etc. Moving forward, employing more realistic experimental environments including real world scenarios and testbeds should become a priority in DTN protocol research.

Some DTN congestion control techniques were designed to operate over specific routing protocols. In a heterogeneous DTN environment where no single routing protocol is universally supported, routing-specific techniques' control over congestion is limited to the nodes where the corresponding routing protocols are in operation. The dependence of congestion control techniques on specific routing protocols can be mitigated by limiting the scope of congestion control information to the node at which that information was generated, i.e., merely using that information as the basis for local admission control. But a more powerful mitigation would be to enable interoperation among congestion control techniques, enabling protocol-neutral network topology information and congestion control cues generated in one routing environment to be propagated to environments governed by other protocols. For example, network regions in which routes are computed from lists of scheduled contacts should be able to make use of congestion control and network topology information developed in regions where routing is opportunistic.

Generality aside, other aspects of the congestion control techniques developed to date seem problematic. Many of these mechanisms function in a reactive fashion and attempt to manage congestion in closed control loops; while the simplicity of closed-loop protocols is appealing, these approaches are innately non-delay-tolerant. Closed-loop mechanisms work very well for local admission control but cannot be relied upon for timely operation over network links that are subject to lengthy lapses in connectivity and/or long signal propagation times.

Accordingly, we suggest that DTN congestion control should employ a hybrid of open-loop and closed-loop control: closed-loop local admission control coupled with open-loop control over network links, functioning both proactively and reactively.

The natural question, then, is whether or not adherence to this principle would enable development of a single universal congestion control mechanism for DTNs. Can any single mechanism be driven by information produced by all possible DTN routing protocols, in all possible operational scenarios (ranging from terrestrial networks to deep space mission operations), maximizing network utilization while minimizing end-to-end delivery latency? These are questions we plan to explore in our future work.

We should also highlight the work being conducted by the IRTF's Delay-Tolerant Networking Research Group (DTNRG) [69], under which a DTN transport-layer overlay is being proposed. This could be an ideal building block upon which a universal, routing-neutral DTN congestion control framework could be built.

Table 6. Flow and Congestion Control Mechanisms

Mechanism	Congestion Detection	Open- or Closed-Loop Control	Proactive or Reactive Control	Application	Contacts	Routing	Evaluation Platform	Deployability
Hop-by-hop local flow control [55]	Network capacity	Closed-loop	Hybrid	Interplanetary Communication	Scheduled	Independent	NS-2	medium
Storage routing for DTN congestion control [43]	Buffer availability	Closed-loop	Reactive	Environmental Monitoring	Predicted and Scheduled	dependent	YACSIM and GT-ITM	medium
Autonomous congestion control [32]	Buffer availability	Closed-loop	Hybrid	Data Dissemination	Predicted	Independent	–	high
Congestion avoidance based on path avoidance [38]	Buffer availability	Closed-loop	Hybrid	Community-based Terrestrial Communication	Predicted	Dependent	OPNET	medium
Context aware forwarding algorithm (CAFÉ) [36] [37]	Buffer availability	Closed-loop	Hybrid	Community-based Terrestrial Communication	Predicted	Dependent	own trace driven opportunistic simulator	medium
Token based congestion control [50]	Network capacity	Closed-loop	Proactive	Environmental Monitoring	Predicted	Independent	own discrete event simulator	medium
Push-Pull custody transfer [42]	Buffer availability	Closed-loop	Reactive	Mobile Terrestrial Communication	Predicted	Independent	DTNSim	medium
Incentive Multi-paths Routing with Alternative Storage (IMRASFC) [46]	Buffer availability	Closed-loop	Hybrid	Mobile Terrestrial Communication	Opportunistic	Dependent	ONE	medium
Distributive congestion control for different types of traffic [33]	Buffer availability	Open-loop	Hybrid	Mobile Terrestrial Communication	Predicted and Opportunistic	Independent	Weka undeployed	high
Congestion Control Routing Algorithm for Security Defense based on Social Psychology and Game Theory (CRSG) [34]	Buffer availability	Closed-loop	Reactive	Community-based Terrestrial Communication	Opportunistic	Independent	ONE	medium
Node-based replication management algorithm (RRCC) [53]	Drop rate	Closed-loop	Hybrid	Mobile Terrestrial Communication	Predicted and Opportunistic	dependent	ONE	medium
Buffer space advertisement to avoid congestion [39]	Buffer availability	Closed-loop	Proactive	Mobile Terrestrial Communication	Opportunistic	Independent	NS-2 and ONE	medium
Congestion aware forwarding algorithm [41]	Buffer availability	Closed-loop	Proactive	Data Dissemination	Predicted	Dependent	ONE	medium
Combined congestion control for IPN [31]	Buffer availability	Closed-loop	Hybrid	Interplanetary Communication	Scheduled	Dependent	NS-2	medium
Threshold-based congestion control protocol [54]	Drop rate	Closed-loop	Reactive	Mobile Terrestrial Communication	Predicted	Dependent	ONE	medium
Lincklider transmission protocol transport (LTP-T) [35]	Buffer availability	Closed-loop	Reactive	Interplanetary Communication	Scheduled	Dependent	network emulation with Netem [70]	medium
Congestion management strategy based on revenue management and dynamic programming [47]	Buffer availability	Open-loop	Proactive	Mobile Terrestrial Communication	Opportunistic	Independent	own discrete event-driven simulator	high
Credit-based congestion control [40]	Buffer availability	Closed-loop	Hybrid	Mobile Terrestrial Communication	Opportunistic	Independent	ONE	high
Average forwarding number based on epidemic routing (AFNER) [51]	Network capacity	Open-loop	Hybrid	Mobile Terrestrial Communication	Opportunistic	Dependent	NS-2	low
Simulated annealing and regional movement (SARM) [45]	Buffer availability	Open-loop	Reactive	Data dissemination	Predicted and Opportunistic	Dependent	ONE	medium
Following routing (FR) [44]	Buffer availability	Open-loop	Reactive	Mobile Terrestre Communication	Opportunistic	Dependent	ONE	medium
Dynamic congestion control based routing (DCCR) [48]	Buffer availability	Closed-loop	Hybrid	Mobile Terrestre Communication	Opportunistic	Dependent	ONE	medium
Message Admission Control based on Rate Estimation (MACRE) [49]	Buffer availability	Closed-loop	Hybrid	Data dissemination	Opportunistic	Dependent	ONE	high
Interplanetary Overlay Network (ION) [68] [52]	Network capacity	Closed-loop	Proactive	Interplanetary Communication	Scheduled	Independent	ION framework	high

6. Conclusion

This paper reviewed the state-of-the-art in DTN congestion control. We started by highlighting how DTN congestion control is different from "traditional" Internet congestion control. We then proposed a DTN congestion control taxonomy which we use to describe existing congestion control mechanisms and place them in context of one another. We anticipate that the proposed taxonomy will help to map the DTN congestion control design space and put in perspective the many existing DTN congestion control techniques. Furthermore, our exploration of the DTN design space will also help to identify important issues and questions that are yet to be addressed.

Our exploration of the DTN congestion control shows that a considerable number of protocols have been proposed, most of which have common goals, namely: increase successful message delivery while decreasing delivery delay and keep network utilization high without congesting it. To achieve these goals, several existing DTN congestion control mechanisms adopt a reactive approach by simply dropping messages stored at DTN nodes to make room for incoming messages. Both "traditional"– and new drop policies, i.e., specifically designed for DTN environments, have been employed. Alternatively, a number of DTN congestion control techniques, instead of discarding messages, try to transfer them to neighboring nodes, using sometimes "back pressure" to adjust message generation rate and admission control to restrict the number of flows entering the network.

We also observe that, in DTNs, cost-performance trade-offs become even more exacerbated. For example, reactive congestion control may have an even more considerable impact on performance because of the inherently high delays: by the time congestion is detected, a large number of messages may have been dropped. Dropping messages is detrimental to performance in general, but even more so in DTNs due to their episodic connectivity and limited resources. When messages are dropped, not only won't they be delivered to their final destination, but they will also have consumed precious network resources. Hybrid approaches, i.e., combining reactive and proactive congestion control strategies, are attractive in DTN environments. Hybrid approaches will try to avoid congestion from happening in the first place but can resort to reactive measures, such as dropping messages, if needed.

Another interesting observation is that several congestion control mechanisms have been proposed to resolve congestion that may result from replication-based message forwarding. As a result, they are interoperable with a variety of routing protocols as long as routing uses message replication as a basis for message forwarding. This is the case of networks that use routing based on epidemic forwarding or a variant thereof (e.g., Prophet [59], Spray-and-Wait [60]). However, these congestion control schemes cannot be classified as routing-independent since they only apply to routing based on message replication. The RRCC mechanism [44] is a notable example as it performs congestion control by limiting the number of message replicas. Even though RRCCs authors consider it to be independent of the routing protocol, according to our classification, RRCC [44] is routing protocol dependent as it would not apply to forwarding-based routing (i.e., routing that does not replicate messages) such as [61], [62] and [63]. One interesting direction for future research is to design effective congestion control mechanisms that can interoperate with any routing scheme.

Traditional Internet congestion control mechanisms typically rely on closed-loop approaches that use end-to-end feedback. However, in DTNs, closed-loop strategies usually employ feedback on a hop-by-hop basis. We contend that DTN congestion control should employ a hybrid of open-loop and closed-loop control so that nodes can also make congestion control decisions based on local information. That way they do not have to rely exclusively on the network to make congestion control decisions.

One interesting observation is that DTN congestion control research relies heavily on simulation platforms to test and evaluate proposed protocols and algorithms. We argue that while simulation-based experimentation is a necessary step, it does not replace real-world experimentation. As such, employing more realistic experimental environments including real world scenarios and testbeds should become a priority in DTN protocol research.

Finally, one important issue from our exploration of existing DTN congestion control techniques is that there is no "universal" congestion control mechanism that will be applicable to all DTN scenarios and applications. To complement our qualitative comparison of DTN congestion control mechanisms, we have been conducting a comparative performance study of different DTN congestion control techniques when applied to different application scenarios. Another future work direction we plan to pursue is to propose novel congestion control mechanisms for DTNs based on our qualitative- and quantitative studies. We will consider Interplanetary Networking scenarios as one of our driving applications.

7. Acknowledgments

Part of the research discussed in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with NASA. This work was partially funded by NSF under project CNS 1321151.

References

[1] Deep space network, http://deepspace.jpl.nasa.gov, jet Propulsion Laboratory, California Institute of Technology. Accessed in February 2012 (2013).
[2] K. Fall, A delay tolerant netwok architecure for challenged internets, in: ACM SIGCOMMM, 2003.
[3] T. Matsuda, T. Takine, (p,q)-epidemic routing for sparsely populated mobile ad hoc networks, Selected Areas in Communications, IEEE (2008) 783–793.
[4] M. Ibrahim, Routing and performance evaluation of disruption tolerant networks, Ph.D. thesis, Université de Nice - Informatique - Sophia Antípolis (November 2008).
[5] T. Spyropoulos, R. N. B. Rais, T. Turletti, K. Obraczka, A. Vasilakas, DTNs: Protocols and Aplications, CRC Press, 2011, Ch. DTN Routing: Taxonomy and Design.
[6] A. Vahdat, D. Becker, Epidemic routing for partially-connected ad hoc networks, Tech. rep., Dept. of Computer Science, Duke University (2000).
[7] K. A. Harras, K. C. Almeroth, E. Belding-royer, Delay tolerant mobile networks (DTMNs): Controlled flooding schemes in sparce mobile networks, in: IFIP Networking, 2005.
[8] A. Lindgren, A. Doria, Probabilistic routing in intermittently connected networks, internet-Draft, draft-irtf-dtnrg-prophet-01 (2008).
[9] A. Lindgren, A. Doria, O. Schelen, Probabilistic routing in intermittently connected networks, in: SIGMOBILE Mobile Computing and Communication Review, Vol. 7, 2003, pp. 19–20.
[10] T. Spyropoulos, K. Psounis, C. S. Raghavendra, An efficient routing scheme for intermittently connected mobile networks, in: 2005 ACM SIGCOMM workshop on Delay-tolerant Networking, 2005, pp. 252–259.
[11] J. Burgess, B. Gallagher, D. Jensen, B. N. Levine, Max-prop: Routing for vehicle-based disruption tolerant networks, in: 25th IEEE Int. Conf. on Computer Communications, 2006, pp. 1–11.
[12] B. Soelistijanto, M. P. Howarth, Transfer reliability and congestion control strategies in opportunistic networks: A survey, IEEE Communications Surveys And Tutorials.
[13] C. Caini, H. Cruickshank, S. Farrell, M. Marchese, Delay and disruption tolerant networking (DTN): An alternative solution for future satellite networking applications, in: IEEE, Vol. 99, 2011, pp. 1980–1997.
[14] I. Psaras, L. Wood, R. Tafazolli, Delay-/disruption-tolerant networking: State of the art and future challenges, Tech. rep., Center for Communication Systems Research, Department of Eletrical Engeneering , University of Surrey (2009).
[15] A. G. Voyiatzis, A survey od delay and disruption tolerant networking applications, Internet Engineering 5 (1).
[16] S. Burleigh, K. Fall, V. Cerf, B. Durst, K. Scoth, H. Weiss, Delay-tolerant networking: An approach to interplanetary internet, IEEE Communication Magazine 41 (6), http://trs-new.jpl.nasa.gov/dspace/handle/2014/40636. Acessed in May 2012.
[17] V. Cerf, S. Burleigh, A. Hooke, L. Targerson, R. Durst, K. Scott, K. Fall, H. Weiss, Delay tolerant networking architecture, internet RFC4838 (April 2007).
[18] K. Scoth, S. Burleigh, Bundle protocol specification, http://www.rfc-editor.org/rfc/rfc5050.txt, request for Comments: RFC 5050 (November 2007).
[19] K. Fall, stephen Farrell, DTN: an architecture retrospective, IEEE Journal on Selected Areas in Communications 26 (5).
[20] V. Jacobson, Congestion avoidance and control, in: A. C. C. Review (Ed.), SIGCOMM, Vol. 18, Stanford, CA, 1988, pp. 314–329.
[21] B. P. Wydrowski, Techniques in internet congestion control, Ph.D. thesis, Electrical and Electronic Engineering Department, The University of Melbourne (February 2003).
[22] R. Srikant, The Mathematics of Internet Congestion Control, Birkuserä Boston, 2004.
[23] Metrics for the evaluation of congestion control mechanism, http://www.ietf.org/mail-archive/web/ietf-announce/current/msg04676.html, internet-DRAFT. Acessed in May 2012 (August 2006).
[24] S. Floyd, Metrics for the evaluation of congestion control mechanisms, http://www.ietf.org/rfc/rfc5166.txt, network Working Group, Accessed in November 06 (March 2008).
[25] L. Mamatas, T. Harks, V. Tsaoussidis, Approaches to congestion control in packet networks, Internet Engineering 1 (1).
[26] S. Farrell, V. Cahill, D. Geraghty, I. Humphreys, When TCP breaks: Delay and disruption-tolerant networking, IEEE Internet Computing 10 (4).
[27] E. Konler, M. Handley, S. Foyd, Datagram congestion control protocol (DCCP), Tech. rep., Network Working Group Request for Comments: 4340, rFC4340 (2006).
[28] S. Ahmad, A. Mustafa, B. Ahmad, A. Bano, A.-S. Hosam, Comparative study of congestion control techniques in high speed networks, International Journal of Computer Science and Information Security 6 (2) (2009) 222–231.
[29] F. Herbertsson, Implementation of a delay-tolerant routing protocol in the network simulator NS-3, Ph.D. thesis, Departament of Computer and Information Science, Linkopings Universitet (22/12/2010).
[30] M. J. Khabbaz, C. M. Assi, W. F. Fawaz, Disruption-tolerant networking: A comprehensive survey on recent developments and persisting challhenges, in: IEEE Communications Survey & Tutorials, 2011.
[31] I. Bisio, M. Cello, T. de Cola, M. Marchese, Combined congestion control and link selection strategies for delay tolerant interplanetary networks, in: GLOBECOM 2009.
[32] S. Burleigh, E. Jennines, J. Schoolcraft, Autonomous congestion control in delay tolerant network, jet Propulsion Laboratory, American Institute of Aeronautics and Astronautic (2007).

[33] A. Chauhan, S. Kumar, V. kumar, S. Mukherjee, C. T. Bhunia, Implementing distributive congestion control in a delay tolerant network for different types of traffic, www.ietf.org/mail.../web/.../docYrH2LrS75z.doc, department of Computer Science and Engineering, Indian School of Mines, Deemed University. Accessed in June 18, 2012 (2012).
[34] W. Cheng-jun, G. Zheng-hu, T. Yong, Z. Zi-wen, Z. Bao-kang, CRSG: A congestion routing algorithm for security defense based on social psychology and game theory in DTN, Journal of Central South University 20 (2) (2013) 440–450, springer.
[35] S. Farrell, A delay and disruption tolerant transport layer protocol, Ph.D. thesis, University of Dublin, Trinity College (September 2008).
[36] A. Grundy, M. RadenKovic, Promoting congestion control in opportunistic networks, in: IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications, 2010, pp. 324–330, department of Computer Science, University of Nottingham.
[37] A. M. Grundy, Congestion framework for delay-tolerant communications, Ph.D. thesis, School of Computer Science, University of Nottingham (July 2012).
[38] D. Hua, X. Du, L. Cao, G. Xu, Y. Qian, A DTN congestion avoidance strategy based on path avoidance, in: 2nd International Conference on Future Computer and Communication, IEEE, Vol. 1, 2010, pp. 855–860.
[39] J. Lakkakorpi, M. Pitkanen, J. Ott, Using buffer space advertisements to avoid congestion in mobile opportunistic DTNs, in: 9th IFIP TC 6 International Conference on Wired/Wireless Internet Communications, 2011, pp. 386–397.
[40] L. Leela-amornsin, H. Esaki, Heuristic congestion control for massage deletion in delay tolerant network, in: Smart Spaces and Next Generation Wired/Wireless Networking, 2010, third Conference on Smart Spaces and 10th International Conference.
[41] M. Radenkovic, A. Grundy, Congestion aware forwarding in delay and social opportunistic networks, in: Eight International Conference on Wireless on Demand Network Systems and Services, 2011, pp. 60–67.
[42] M. Seligman, K. Fall, P. Mundur, Alternative custodians for congestion in delay tolerant networks, in: SIGCOMM'06 Workshops, 2006.
[43] M. Seligman, K. Fall, P. Mundur, Storage routing for DTN congestion control, in: Wireless Communications and Mobile Computing, Vol. 7, 2007, pp. 1183–1196.
[44] C. Wang, B. Zhao, W. Peng, C. Wu, Z. Gong, Following routing: An active congestion control approach for delay tolerant networks, in: 2012 15Th International Conference on Network-Based Information Systems, 2012, pp. 727–732.
[45] C. Wang, B. Zhao, W. Yu, C. Wu, Z. Gong, SARM: An congestion control algorithm for dtn, in: 9th International Conference on Ubiquitous Intelligence and Computing and 9th International Conference on Autonomic and Trusted Computing, IEEE Computer Societ, 2012, pp. 869–875.
[46] L. Yin, hui-mei lu, Y. da Cao, J. min Gao, An incentive congestion control strategy for DTNs with mal-behaving nodes, 2010, pp. 91–94.
[47] G. Zhang, Y. Liu, Congestion management in delay tolerant networks, in: WICON, 2008, pp. 1–9.
[48] S. C. Lo, C. L. Lu, A dynamic congestion control based routing for delay tolerant network, in: 9th International Conference on Fuzzy Systems and Knowledge Discovery, 2012, pp. 2047–2051.
[49] Y. An, X. Luo, MACRE: A novel distributed congestion control algorithm in DTN, Trans Tech Publications, Advanced Engineering Forum 1 (2011) 71–75.
[50] E. Coe, C. Rachavendra, Token based congestion control for DTN, in: Aerospace Conference, IEEE, 2010.
[51] L. Yun, C. Xinjian, L. Qilie, Y. Xianohu, A novel congestion control strategy in delay tolerant networks, in: Second International Conference on Future Networks, 2010.
[52] S. Burleigh, Interplanetary overlay network - design and operation, Tech. Rep. JPL D-48259, Jet Propulsion Laboratory/ California Institute of Technology (2013).
[53] N. Thompson, S. C. Nelson, M. Baknt, T. Abdelzaher, R. Kravets, Retiring replicants: Congestion control for intermittently - connected networks, in: INFOCOM'2010, IEEE, 2010, pp. 1–9, university of Illinois at Urbana-Champaing, Department Of Computer Science.
[54] N. Thompson, R. Kravets, Understanding and controlling congestion in DTNs, in: Mobile Computing and Communications, Vol. 13, 2010.
[55] F. D. Rango, M. Tropea, G. B. Laratta, S. Marano, Hop-by-hop local flow control over interplanetary networks based on DTN architecture, in: Communications, 2008. ICC'08. IEEE International Conference on, 2008, pp. 1920–1924.
[56] J. A. Davis, A. H. Fagg, B. N. Levine, Wearable computers as packet transport mechanism in highly-partitioned ad-hoc networks, in: International Symposium on Werable Computing, Zurich, 2001, pp. 141–148.
[57] S. Rashid, Q. Ayub, Effective buffer management policy DLA for DTN routing protocols under congestion, international Journal of computer and Network Security 2 (9) (2010) 118–121.
[58] A. Indgren, K. S. Phanse, Evaluation of queuing policies and fowarding strategies for routing in intermittently connected networks, in: IEEE COMSWARE, 2006, pp. 1–10.
[59] A. Krifa, C. Barakat, T. Spyropoulos, Optimal buffer management policies for delay tolerant networks, 2008.
[60] Q. Ayub, S. Rashid, T-drop: An optimal buffer management policy to improve QoS in DTN routing protocols, Journal of Computing 2 (10) (2010) 46–50.
[61] S. Rashid, Q. Ayub, M. S. M. Zahid, S. H. Abdullah, E-DROP: An effective drop buffer management policy for DTN routing protocols, International Journal of Computer Applications 13 (7).
[62] Q. Ayub, S. Rashid, M. S. M. Zahid, Buffer scheduling policy for opportunistic networks, International Journal of Scientific & Engineering Research 2 (7).
[63] S. Rashid, A. H. Abdullah, M. S. M. Zahid, Q. Ayub, Mean drop and effectual buffer management policy for delay tolerant network, European Journal of Scientific Research 70 (3) (2012) 396–407.
[64] D. Kim, H. Park, I. Yeom, Minimize the impact of buffer overflow in DTN, in: International Conference on Future Internet Technologies, 2008.
[65] Y. Li, M. Qian, D. Jin, L. Su, L. Zeng, Adaptive optimal buffer management policies for realistic DTN, in: IEEE GLOBECOM 2009, 2009.
[66] The network simulator ns2, http://nsnam.isi.edu/nsnam/index.php/User_Information, accessed in March 2013 (2013).
[67] The ONE, http://www.netlab.tkk.fi/tutkimus/dtn/theone/, accessed in March 2013 (2013).
[68] J. P. Laboratory, ION: Interplanetary overlay network, https://ion.ocp.ohiou.edu/, accessed in January 2013 (2013).
[69] Delay-tolerant networking research group, https://sites.google.com/site/dtnresgroup/home (July 2014).
[70] S. Hemminger, Network emulation with NetEm, Linux Conference Australia - LCA2005 (April 2005).