

A MULTI-CUSTODIANS DISTRIBUTED STORAGE MECHANISM FOR DTN STORAGE-BASED CONGESTION PROBLEM

Ching-Neng Lai

Department of Information Technology
Hsing Wu University
24452 Taipei City
Taiwan (R.O.C.)

Email: 093062@mail.hwu.edu.tw
Tel.: +886-2-2601-5310
Fax: +886-2-2601-0748

ABSTRACT

Nowadays, distributed storage is adopted to alleviate Delay-tolerant networking (DTN) congestion, but reliability transmission during the congestion period remains an issue. In this paper, we propose a multi-custodians distributed storage (MCDS) framework that includes a set of algorithms to determine when should appropriate bundles be migrated (or retrieved) to (or from) the suitable custodians, so that we can solve DTN congestion and improve reliability transmission simultaneously. MCDS adopts multiple custodians to temporarily store the duplications of migrated bundle for releasing DTN congestion. Thus, MCDS has more opportunities to retrieve the migrated bundles when network congestion is mitigated. Two performance metrics are used to evaluate the simulation results: the goodput ratio (GR) that represents QoS of data transmission, and the retrieved loss ratio (RLR) that reflects the performance of reliability transmission. We also use another distributed storage mechanism based on single-custodian distributed storage (SCDS) to evaluate MCDS. Simulation results show that MCDS has better GR and RLR in almost all simulation cases. For various scenarios, the GR and RLR of MCDS are in the range of 10.6%-18.4% and 23.2%-36.8%, respectively, which are higher than those of SCDS.

Keywords: *delay-tolerant networking, distributed storage, storage-based congestion control, transfer reliability.*

1.0 INTRODUCTION

Delay-tolerant networking (DTN) is an architecture that attempts to enable the communication over Intermittently-Connected Networks (ICNs). As shown in Fig. 1, DTN introduces a new layer, called Bundle layer, placed between Application and Transport layers. This layer supports store-and-forward and hop-by-hop messages exchanging between intermediate nodes rather than between only the source and destination nodes. Through this mechanism, DTN can minimize the impact of certain network environments which are unable to provide end-to-end reliability and short delays, and further allow existing TCP/IP-based applications to operate unmodified [1, 2]. Therefore, DTN is widely applied to a broad class of network environments with intermittent connectivity or long and varying delays [3, 4], such as interplanetary internets [5, 6], underwater sensor networks [7], vehicular networks [8], and ZebraNet [9].

DTN provides the functionality called custody transfer, the store-and-forward hop-by-hop transport in DTN, to guarantees reliable transmission. The basic idea of custody transfer is that the source node will delegate the responsibility of delivery and retransmission to the next hop node when ensuring that this hop node has successfully received the entire data. Once this node accepts the delegation request, it becomes the “custodian” of the associated data and takes the delivery responsibility. After making sure these data are successfully and completely delivered to the next hop node by the custodian, the custodian will transfer its custody responsibility to its next hop [1, 2]. This procedure repeats until the destination node receives the entire sending data. It should be noted that the routing in DTN is based on flooding algorithm. This is because that disconnection may suddenly happen, yielding that packets in buffer do not have enough time to determine the routing and/or another opportunity to forward [10]. Therefore, a DTN node would frequently receive flooding data and takes custody of multiple data transmissions, resulting in the requirement of a large buffer size [11].

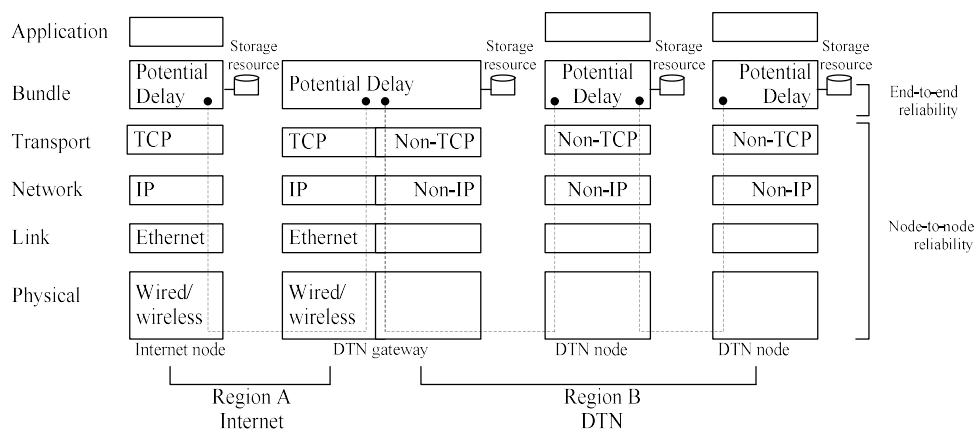


Fig. 1. The architecture of delay tolerant network

However, the buffer size of a DTN node, especially in mobile network, is limited to accommodate. It would have the custody of flooding data from multiple sources for long time [10]. If a custodian cannot properly reject some new custody requests or offload its stored data to other nodes in time, its storage resource will be eventually exhausted. This causes the prohibition of future traffic incoming even when some outgoing paths are still available [12]. This situation is referred to as storage-based congestion. In order to cope with this problem, current DTN suggests rejecting new custody requests when the storage space is unavailable or discarding the expired messages stored in the buffer [1]. Nevertheless, rejecting custody requests is not practical for the upstream node (i.e., sensor) which has small buffer size and produces data continuously. On the other hand, discarding expired data stored in the queue will be undesirable if the dropped data are extremely important and the upstream source of the dropped data has no alternation to be delivered. Therefore, both approaches may damage the reliability of message delivery for uncontrollable upstream sources.

Look after both sides of transfer reliability and congestion control strategies are the significant topics of DTN [13]. Instead of simply rejecting new custody requests or discarding stored data as aforementioned, we investigate other alternative methods that solve the DTN storage-based congestion problem as follows. Traditional congestion control mechanisms, such as the methods applied in TCP [14], SCTP [15] and DCCP [16], rely on an uninterrupted path to timely relay congestion information. However, DTN cannot provide such a stable path to inform senders in a timely base, and thus the associated strategies are not applicable. Therefore, some researches proposed storage-based congestion control methods to mitigate the DTN storage-based congestion problem [11, 12, 17]. The basic idea is that a custodian temporarily distributes its data to some of its neighbor nodes with unused storage resource when encountering storage-based congestion. Later on, when the congestion situation is relieved, the custodian will retrieve the data from the associated neighbor nodes. This strategy can temporarily acquire more storage resource for new custody requests without discarding any important stored data. It can also help keep the traffic flow from the upstream sources equipped with limited storage resource and weak connection ability. Hence, reliability of DTN message delivery is better ensured. However, the methods proposed in [11, 12, 17] only choose one neighbor node having sufficient storage resource to temporarily keep the data but do not consider the accessibility of this node when the custodian wants to retrieve the data. More specifically, placing the data in only one neighbor node would yield a high risk of retrieval failure because the custodian may lose connection with this node anytime.

In light of the limitations of the current DTN storage-based congestion control methods, we propose a Multi-Custodians Distributed Storage (MCDS) method. MCDS enables the congestion custodian to temporarily place its duplication of migrated bundle in several custodians in order to reduce the risk of losing retrieving the migrated bundle. MCDS consists of the following steps: (1) when the custodian detects storage-based congestion, MCDS chooses the bundle having less or no chance to be forwarded as the migrated bundle (e.g., because the next hop of the bundle has low or zero accessibility) and then (2) selects the custodians not verging on congestion and have higher accessibility as the Temporary Storage Resources Custodian (TSRC). (3) MCDS duplicates the migrated bundle and transmits the duplications to multiple TSRCs. When the ACK messages from TSRC are received, the migrated bundle is removed from the congested custodian. (4) Once the network congestion is relieved, MCDS retrieves the migrated bundle from one of the TSRCs. MCDS can solve the problem of DTN congestion by

increasing the opportunities of successful retrieval of the migrated bundles, which effectively improves the QoS of reliability transmission in DTN.

According to the characteristics of MCDS, some existing DTN networks such as the Vehicular Network (that is applied to telematics) and the ZebraNet (developed for monitoring the long-term behaviors of wild animals) are examples that can adopt this mechanism for improving the data transmission performance. When these clustering DTN networks are congested, even though the custodians that carry the data are out of the transmission range, MCDS could still provide reliable transmission. To address the problem of DTN congestion and improve the transmission reliability, MCDS has the following tradeoffs. (1) When the network is congested, the bundles would be temporarily placed in multiple neighbor nodes, and thus the node should provide storage resources for the whole network. (2) For reliability transmission, MCDS should spend a little extra transmission time in the implementation. Therefore, the data source that is delay tolerable is suitable to be transmitted with MCDS.

This paper is organized as follows: Section 2 reviews DTN and the present congestion control mechanisms of traditional Internet. Section 3 thoroughly introduces the underlying principles of MCDS and the associated operations. Simulation results and comparisons to other related methods are presented in section 4. Finally, conclusions are given in Section 5.

2.0 RELATED WORKS

To cope with DTN's storage-based congestion problem, existing approaches can be classified into two types [18]: passive methods and initiative methods. Passive methods are based mainly on access control, where custodians drop bundles or cease accepting bundles when the network is congested. In [10], Yun et. al propose an average forwarding number based epidemic routing (AFNER) method to alleviate the congestion network. AFNER computes the average forwarding number of all of the packets from the sending to the receiving custodians as a reference value. When congestion occurs, AFNER will drop the packets whose forwarding number is bigger than the reference value to release the storage resource. In [19], Bisio et. al adopt the DTN architecture on the interplanetary network (IPN). The message bundles of IPN are classified as "expendable" if they have the shortest delivery delay, and as "normal" if they have to follow strict reliability constraints. The bundles acquire different dropping priorities according to the classifications, where the dropping priority of "expendable" is higher than that of "normal." The buffer management strategy of custodians is Random Early Detection (RED). When the network congestion occurs, the custodians drop bundles via RED and the bundle priority. Beside in [20], Yang et. al propose a congestion avoidance strategy which adopts both buffer management policy and packet receiving policy to execute access control when DTN is congested.

Initiative methods handle the congestion problem via custody transferring and storage resources management. These methods focus on the operations of determining what bundles should be migrated to a suitable custodian and when to perform the migration. The approach in [21] adds explicit application hints to the bundles with those hints visible to every custodian. The custodians can cache content according to the hints for application-specific forwarding or act as distributed storage. In [11, 17], several algorithms are proposed to solve the problem of congestion DTN. The authors use a linear combination of the size of available storage resources, link latency and link bandwidth to define an evaluation metric for temporary storage custodian selection. When the congestion situation is mitigated, the retrieval process follows an opposite order of the previous operations that forward bundles from the congestion custodian to the temporary storage custodian. In [12, 22], the authors suggest that the factors of evaluation metric for temporary storage custodian selection should be assigned with the weights according to the application of DTN. Besides, some custodians of DTN could be noted as mal-behaving custodians that are reluctant to store and forward the bundles. In [23], a mechanism is proposed to increase the incentive of normal custodians that are willing to store and forward bundles for other custodians (including the mal-behaving ones). When the mal-behaving custodians get the responses, their willingness to serve other custodians will be increased when the network becomes congested.

In the researches aforementioned, options for mitigating the storage-based congestion include (1) dropping expired bundles, (2) rejecting custody delegation request, (3) lowering the traffic entry in the DTN network, and (4) temporarily moving bundles to somewhere else. Current DTN standard suggests (1) or (2) [1]. Nevertheless, as discussed in Section 1, discarding expired data will be undesirable if the dropped data are important or the upstream source has no alternation to re-deliver these data. With regard to (3), it requires an uninterrupted connection path between the sender and receiver to exchange the traffic congestion information, such as the conventional congestion control mechanism of TCP, SCTP and DCCP. However, DTN cannot provide such a stable path in a timely base, and thus this option is not necessarily applicable.

Although the distributed storage congestion control mechanism of DTN can moderate the congestion situation efficiently, there still remain some issues in the solutions:

- (1) Reliability transmission cannot be guaranteed. In general, the custodians of DTN are usually motional, causing the links volatile and intermittently disconnected. If the custodian storing the migrated bundles cannot connect to the custodian that originally owns these bundles, they cannot be retrieved when needed or could even be lost forever. No matter whether the migrated bundle can be retrieved successfully, the main issue lies in how to increase the accessibility between custodians so that more opportunities can be attained for migrating or retrieving bundles.
- (2) Besides, in previous studies, the selection process of temporary storage custodian does not consider the problem that the custodian could be on the verge of congestion situation, although it still has the capacity to store the migrated bundles. If a custodian of this kind is selected, on the contrary, another storage-based congestion problem will be caused.

3.0 MCDS FOR STORAGE-BASED CONGESTION CONTROL PROBLEM

3.1 Principle of MCDS

How to solve storage-based congestion control and improve the reliability of transmission simultaneously are significant issues in DTN. In this paper, we introduce multi-custodians distributed storage (MCDS), which is a new storage-based congestion control mechanism for DTN. When the network is congested, MCDS duplicates the bundles which should be migrated for releasing the network congestion, and transmits these duplicated bundles to multiple TSRCs. Adopting multiple custodians as storage resources is effective to increase the opportunity of retrieving migration bundles and improve the performance for reliability transmission. To be specific, we define four kinds of custodians in the following: Sending Custodian (SC), Receiving Custodian (RC), Temporary Storage Resources Custodian (TSRC) and Adjacent Custodian (AC). In this paper, we assume RC plays the role as the congestion custodian. The environment of DTN is changing constantly, and MCDS can adaptively detect the timing for executing the congestion control mechanism. The operations of MCDS are divided into three phases described below (Fig. 2):

Phase 1: RC receives data from SC and collects the information from ACs when DTN is in the normal state.

Phase 2: MCDS uses the collected information to identify whether RC is verging to the congestion state.

Phase 3: Once the storage-based congestion is detected, congestion control mechanism is triggered. RC selects and duplicates the appropriate bundle that will be migrated to multiple TSRCs. When the congestion is alleviated, RC retrieves the appropriate bundle from one of the suitable TSRCs. While the bundle is retrieved, the duplications of the migrated bundles stored in the other TSRCs should be removed.

Because the selected TSRCs have high accessibility to RC, when the congestion situation is disencumbered, RC has more opportunities to retrieve the duplicated migrated bundle.

Fig. 3 shows the flow chart of MCDS. To accomplish the three phases of congestion control mechanism, there are four modules designed for MCDS. As shown in Fig. 3, MCDS consists of four modules: adjacent custodian information collection module (ACICM), buffer management module (BMM), TSRC module (TSRCM) and bundle selection module (BSM).

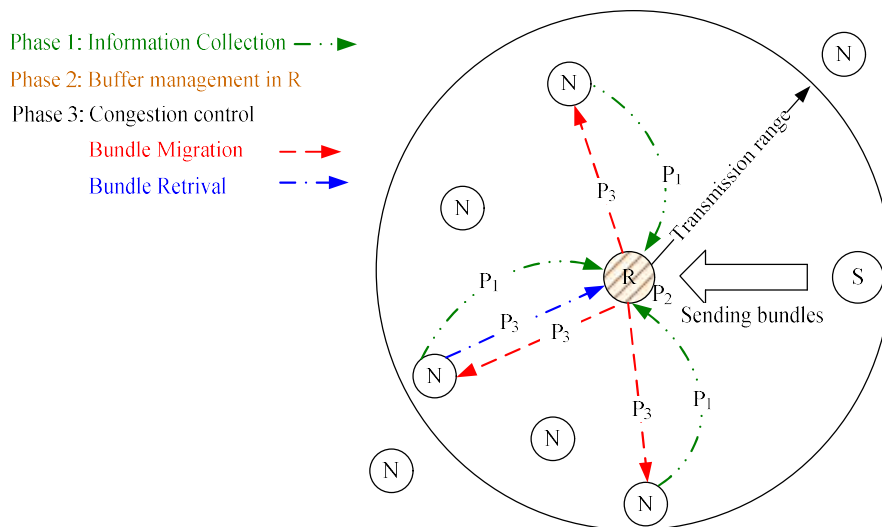


Fig. 2. Operations of congestion control between custodians in MDCS

In order to grasp the status of ACs, MCDS uses ACICM to collect the information from ACs frequently. The storage-based congestion problem of DTN results from the excessive quantity of data (QoD) in the storage resource. In order to monitor the status of custodian, BMM uses the information from ACICM and the variation of QoD to detect whether a custodian is in the congestion state. If the custodian is in congestion, MCDS should select a bundle to be migrated in the congestion custodian and determine the target TSRCs for migration. Thus, MCDS not only uses TSRCM to calculate an evaluation index for picking out TSRCs from ACs but also uses BSM to decide which bundle should be migrated or retrieved. Details are given as follows.

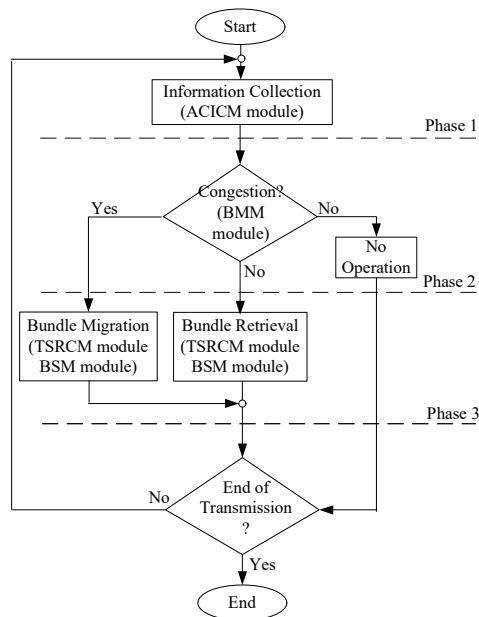


Fig. 3. The flow chart of MCDS

3.2. Phase 1: Adjacent Custodian Information Collection

Because the deployment of custodians and topology of DTN are varied frequently, it is possible that the custodians connect at present but disconnect later. Because MCDS needs to cooperate with the other custodians, the information collected from ACs is necessary. There are four parameters influencing the operation of MCDS significantly: (1) the capacity of storage resources (SR), (2) round trip time (RTT), (3) degree of congestion (DC) of the custodians and (4) accessibility ratio of custodians (λ). To obtain the information from ACs, MCDS needs to broadcast a self-defined probe message to ACs frequently, and then ACs will reply it. The format of probe message is shown in Fig. 4. The fields include the sequence number of the probe message (SN), ID of source custodian (SID), ID of destination custodian (DID), time stamp (TS), SR and DC. A specific parameter, DC, is designed in the probe message. DC is used to indicate the ratio of data occupying the storage resources. The reason is that even when an AC has sufficient storage resources, it could still be oncoming exhausted. Thus, whether the AC is suitable to be a TSRC still becomes uncertain. In the message format, a larger DC means that the AC is closer to the congestion state.

SN	SID	DID	SR	DC	TS
----	-----	-----	----	----	----

Fig. 4. The format of the probe message

Pseudo codes of ACICM are given in Fig. 5. In the information collected, DC and SR can be directly acquired from the ACK message but RTT and λ should be estimated. To evaluate RTT, TS and the time of receiving the ACK messages are used. More specifically, when RC receives the i -th Ack message (ACK_i^k), the RTT between RC and AC_k (denoted as RTT_i^k) can be measured, and the estimated RTT at this instant (denoted as RTT_i^k) is obtained by using an exponential smoothing method in Eq. (1):

$$RTT_{i+1}^k = \frac{1}{8}RTT_i^k + \frac{7}{8}RTT_i^k \quad (1)$$

The accessibility ratio λ is obtained according to Np and Na_i^k , where Np is the number of times that the probe messages are sent by RC and Na_i^k is the number of accumulative ACK_i^k received by the RC. The successful accessibility (λ_i^k) between the RC and the AC_k is calculated by Eq. (2), and then λ_{i+1}^k is obtained by using an exponential smoothing method in Eq. (3).

$$\lambda_i^k = \frac{Na_i^k}{Np} \quad (2)$$

$$\lambda_{i+1}^k = \frac{1}{8}\lambda_i^k + \frac{7}{8}\lambda_i^k \quad (3)$$

Hereafter, for convenience, all the collected information of ACs are presented as the set form, $\mathbf{SR}:\{SR_i^1, SR_i^2, \dots, SR_i^k\}$, $\mathbf{RTT}:\{RTT_i^1, RTT_i^2, \dots, RTT_i^k\}$, $\mathbf{DC}:\{DC_i^1, DC_i^2, \dots, DC_i^k\}$ and $\lambda:\{\lambda_i^1, \lambda_i^2, \dots, \lambda_i^k\}$.

```

ACICM() {
    set Probei message fields;
    do{
        broadcast Probei message;
    }while (Ack message received==false && broadcasting time > RTTmax)

    If( Ackik message received==TURE) /*Ack message from custodian k */
    {
        measure RTTik ;
        Np = times of the probe messages sent;
        Naik = times of accumulative acknowledge;

        calculate  $\lambda_i^k = \frac{Na_i^k}{N_p}$  ;
        RTTi+1k =  $\frac{1}{8}RTT_i^k + \frac{7}{8}RTT_i^k$ ;
         $\lambda_{i+1}^k = \frac{1}{8}\lambda_i^k + \frac{7}{8}\lambda_i^k$ ;
        store collected information in RC;
    }
}
    
```

Fig. 5. The pseudo codes of ACICM

3.3 Phase 2: Buffer management

To monitor the network topology and the QoD variation in the storage resources, MCDS adopts BMM to determine the timing for triggering the congestion control mechanism. BMM divides the storage resources into three parts: safety region, cushion region and hazard region. Ranges of these regions are adaptively adjusted to fit the changeful environment of DTN. If the QoD is in the hazard region, the storage resources are about to be exhausted. The appropriate bundle should then be migrated and its duplications are temporarily stored in TSRCs. On the contrary, if the QoD is in the safety region, the duplications of bundle that are migrated before could be retrieved from the TSRCs.

If we divide the storage resources into only hazard and safety regions, RC may encounter the Ping-Pong Effect. That is, the custodian would keep migrating and retrieving bundles when the QoD is nearby the critical point of the hazard region. This is the reason why we design a cushion region to avoid such Ping-Pong Effects. The regions of storage sources are shown in Fig. 6, where points A and B represent the critical points of hazard region and safety region, respectively.

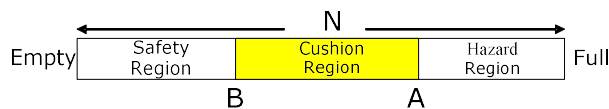


Fig. 6. The state regions of storage resources

How to set the capacity of each region depends on the operation types of MCDS. For instance, the operation of bundle migration is executed when the QoD is in the hazard region. To ensure that the storage resources of RC are not exhausted during the bundle migration period, a sufficiently large hazard region is needed for absorbing the increased QoD. Besides, the RTTs between RC and ACs (denoted as RTT_{RA}) are also significant for determining the time of triggering the bundle migration. For example, if RC cooperates with some ACs whose RTT_{RA} are short, RC can start executing the bundle migration later because less time will be spent for migrating the bundles. According to the above rational, RTTs can help determine the triggering time of bundle migration. Thus, in our system, the capacity of hazard regions is estimated by RTTs, which will be detailed below.

We denote RTT_{RA-max} to be the maximum RTT_{RA} required to accomplish a bundle migration. Hence, when RC receives an ACK message from an AC with RTT_{RA-max} , the ACK messages which are sent from the other ACs can be assuredly received. We use RTT_{RA-max} for estimating the capacity of hazard region, the critical point A is calculated by Eq. (4).

$$A = N - \Delta r \cdot RTT_{RA-max} \tag{4}$$

In Eq. (4), N is the capacity of storage resources (as shown in Fig. 6), Δr is the data variation rate between

incoming and outgoing data. The hazard region is obtained by $\Delta r \cdot RTT_{RA-max}$. As soon as the QoD is over the critical point A, the operation of bundle migration is triggered.

The reason to design the cushion region is to avoid the aforementioned Ping-Pong Effect. When RC is not in the hazard region, there is still no need to initiate the migration of bundles actively. Beside, the data consumption of RC is passively because it is depended on whether the available custodians are nearby RC. In this situation, SC plays a key role to influence the variation of QoD because SC could determine to transmit data to RC at any time. From this point of view, the RTT between SC and RC (RTT_{SR}) is used to estimate the capacity of cushion region. The critical point B (associated with the cushion region as shown in Fig. 6) is calculated by Eq. (5). When QoD is in the cushion region, MCDS can execute the operations of bundle migration or retrieval, depending on a probability proportional to the ratio of QoD that occupies the storage resource. A higher ratio reveals a higher opportunity to migrate bundles than to retrieve bundles. Once the critical points A and B are decided, the rest part of storage resources becomes the safety region.

$$\begin{aligned}
 B &= N - (\text{QoD of Hazard region} + \text{QoD of Cushion Region}) \\
 \text{QoD of Hazard region} &= \Delta r \cdot RTT_{RA-max} \\
 \text{QoD of Cushion region} &= \Delta r \cdot RTT_{SR} \\
 B &= N - \Delta r \cdot (RTT_{RA-max} + RTT_{SR}) \tag{5}
 \end{aligned}$$

There is another evaluation index, DC, revealing the degree of congestion of the custodian. Despite the storage resources of AC are sufficient for the incoming bundle, they would probably verge on a congestion state, and AC of this kind is not good enough as being the TSRC. DC is defined as the ratio of QoD occupying the hazard region in Eq. (6). If the value is negative, we simply set DC as zero because the status is in the safe state. The pseudo codes of BM module are presented in Fig. 7.

$$DC = \max\{(QoD - (N - \Delta r \cdot RTT_{RA-max})) / (\Delta r \cdot RTT_{RA-max}), 0\} \tag{6}$$

```

Operation BM() {
    N = capacity of storage resources;
    Δr = data variation rate;
    RTTmax = Max(quicksort(RTT));
    A = N - Δr · RTTRA-max;
    RTTSR = Round Trip Time between RC and SC;
    B = N - Δr · (RTTRA-max + RTTSR);
    QoD = quantity of data in the storage resources;
    DC = (QoD - (N - Δr · RTTRA-max)) / (Δr · RTTmax);
    Store in MAP;
    if (QoD > A) //hazard region
    {
        return Migration;
    }
    else (QoD > B && QoD < A) //cushion region
    {
        result = probability calculation;
        if (result == migrate)
            return Migration;
        else
            return Retrieval;
    }
    else //safety region
    {
        return RetrievalOrNOP;
    }
}
    
```

Fig. 7. The pseudo codes of BM module

3.4 Phase 3: Storage-Based Congestion control

For MCDS, we design the TSRC module (TSRCM) to select the optimal ACs as TSRCs. We also design the bundle selection module (BSM) to select appropriate bundles for migration or retrieval. Details of TSRCM and BSM are described as follows.

3.4.1 TSRCM

TSRCs are selected to store the duplicated migrated bundles temporarily. Hence, the ACs whose storage resources are sufficient for storing the duplicated bundles should be sieved out as the candidate TSRCs. Besides, to select the optimal TSRCs from these candidates, TSRCM uses a linear combination of exponential functions to form an evaluation index for each AC, as shown Eq. (7),

$$Eval_i^k = \ln(e^{1-DC_i^k} + e^{\lambda_i^k} + e^{\frac{SR_i^k}{SR_{max}}} + e^{\frac{RTT_{min}}{RTT_i^k}}) \quad (7)$$

In Eq. (7), a higher *Eval* value indicates that the custodian has a higher potential to become TSRCs. The powers of the exponential terms are normalized to the interval(0, 1) to balance the contributions of these terms. Because *DC* represents the degree of congestion for the custodian, a lower *DC* reflects a higher evaluation value. Thus, $1 - DC_i^k$ is used in the first term of Eq. (7). In the second term, λ is a positive value less than 1, indicating the degree of the robust connectivity between the custodians. In the third term, the largest SR ($SR_{max} = \max\{\mathbf{SR}\}$) is used as the denominator, which makes the exponent between 0 and 1. Finally, the relationship between *Eval* and RTT in the fourth term is opposite to that of SR, where the priority of custodian with a shorter RTT is higher than that with a longer RTT. Thus, the smallest RTT ($RTT_{min} = \min\{\mathbf{RTT}\}$) is used as the numerator and the other RTTs as the denominators for the index of the fourth term.

When the congestion status is solved, RC does not require retrieving the migrated bundle immediately because DTN is delay tolerant. The migrated bundle should only be retrieved during the period satisfying the requirement of DTN's applications. Consider the probability density function (PDF) and the cumulative distribution function (CDF) of Poisson Distribution in Eqs. (8) and (9), respectively. TSRCM can then calculate the reliably accessed probability that an AC can successfully connect to RC for more than n times in the period T .

$$P_i^k(m) = \frac{(\lambda_i^k T)^m}{m!} e^{-(\lambda_i^k T)} \quad (8)$$

$$P_i^k(m > n) = 1 - \sum_{k=0}^n \frac{(\lambda_i^k T)^m}{m!} e^{-(\lambda_i^k T)} \quad (9)$$

Finally, to ensure that the migrated bundle can be retrieved from ACs, the reliability analysis of the parallel system is adopted as shown in Eq. (10), the X is defined as the threshold ratio of reliably accessible probability. In accordance with the $Eval_i^k$ of ACs in Eq. (7), from the higher to lower, TSRCM substitutes the results of Eq. (9) into Eq. (10) sequentially, until X is equal to or exceeds a specific value (such as $X \geq 95\%$). Then the ACs whose accessibilities are substituted into Eq. (10) are chosen as TSRCs. The pseudo codes of TSRCM are shown in Fig. 8.

$$X = 1 - \prod_{i=1}^j (1 - P_i^k(m > n)) \quad (10)$$

```

TSRS(){
  while( ACKik.SR ≥ size of incoming bundle) /* SR : storage resource */
  {
    candidatem = ACij;
    m ++; }
    SRmax = MAX(quicksort(candidatem.SR));
    RTTmin = MIN(quicksort(candidatem.RTT));
    for (j = 0; j <= m; j ++ )
    {
      Evalj = ln(e $\frac{\text{candidate}_j.\text{SR}}{\text{SR}_{\text{max}}}$  + e1 - candidatej.DC + e $\frac{\text{RTT}_{\text{min}}}{\text{candidate}_j.\text{RTT}}$  + ecandidatej.λ);
      candidatej.Eval = Evalj;
      Pj =  $\frac{(\text{candidate}_j.\lambda T)^k}{k!} e^{-(\text{candidate}_j.\lambda T)}$ 
      Pj (k > n) =  $1 - \sum_{k=0}^n \frac{(\text{candidate}_j.\lambda T)^k}{k!} e^{-(\text{candidate}_j.\lambda T)}$ 
      candidatej.p = Pj (k > n); }
      quicksort(candidatej.EST);
      do { temp = temp * candidatej.p;
          selected[] = candidatej;
        } while(1 - temp ≥ X);
    }
}

```

Fig. 8. The pseudo codes of TSRCs module

3.4.2 BSM

The pseudo codes of BSM are shown in Fig. 9. The principle we adopt for bundle selection is that the bundle of a less chance to be forwarded to the next hop custodian will have a higher priority to be selected for migration. There are two rules for selecting the migrated bundles. First, the bundles whose next hop custodians have lower accessibilities to RC are selected, because bundles of this kind will occupy the storage resources for a long time. Second, the bundles in the tail of the storage resources are selected because they will not be immediately forwarded. In these two cases, the bundles could waste the storage resource when the network is congested.

When the migrated bundle is selected, RC transmits the duplications of the migrated bundle to TSRCs. When the duplicated bundles are received, TSRCs transmit the Ack messages to RC. Once RC receives the Ack messages, it removes the migrated bundle from the storage resources. A mapping is established to record the relationship between the duplicated bundles and TSRCs.

When the congestion state is alleviated, according to the mapping information, BSM can determine the TSRC that should be retrieved for a duplicated bundle. For the bundle retrieval operation, the duplicated bundle whose next-hop custodian is in the available transmission range of RC has a high priority to be retrieved. Once a TSRC having the required bundle appears in the communication range, RC will send a retrieval request message to that target TSRC. Then, the target TSRC transmits the duplicated bundle to RC. When RC retrieves the required bundle, the Ack messages are sent to the TSRCs that store the same duplicated bundles, and these TSRCs will drop the duplicated bundles when receiving the Ack messages.

```

BS( operation op, array_of_custodian selected[] ) {
    find=false;
    AC[]={custodians in the communication range of RC};
    if( op == Migration ){
        candidate_bundle=MIN(quicksort(bundle,nextthop by λ);
        do { if(candidate_bundle.nextthop not in AC[])
            migration_bundle=candidate_bundle;
            else migration_bundle= tail bundle of storage resources;
            do{ transmits duplicated migration_bundle to ACs in selected[];
                }while(one of Ack message received);
                remove bundle;
            }while(congestion==false);
    }elseif( op == Retrieval ){
        find = false;
        candidate_bundle[] =quicksort(migrated_bundle by playout_time from low to high);
        while (not end of candidate_bundle[]){ if( candidate_bundle[i] in AC[])
            { retrieval_bundle=candidate_bundle[i];
              find=true;
              break;}
            else i++; }
        if (find==false){ candidate_bundle[] =quicksort(migrated_bundle,nextthop by λ from high to low);
        while (not end of candidate_bundle[]){ if( candidate_bundle[i] in AC[])
            { retrieval_bundle= candidate_bundle[i];
              find=true;
              break;}
            else i++; }}
        target_custodian[] = search (custodians with retrieval_bundle in AC[]);
        send retrieval message to target_custodian[];
        if( retrieval_bundle received )
            send remove duplicated bundle message to target_custodian[];
        }}
}

```

Fig. 9. The pseudo codes of BS module

4.0 PERFORMANCE EVALUATION

To demonstrate that MCDS not only releases congestion status of DTN effectively but also keeps data reliably transmitted during the network congestion period, a DTN simulation environment is constructed for evaluation. To tackle most of the situations possibly happen in DTN, four scenarios are proposed. These scenarios are based on different simulation environments of (1) the number of custodians, (2) the size of storage resources, (3) the threshold ratio of reliable accessibility, and (4) the moving rate of the custodians. In this section, we first describe the details of simulation environment in Section 4.1, follow by the simulation results and analyses.-

4.1 Environment setup

There are some reasons resulting in the bundle loss when the network is congested: (1) No TSRCs are available for migrating the bundles, and thus the congestion custodian drops the bundles. (2) When the congestion status is released, the previous congestion custodian cannot retrieve the migrated bundle because the associated TSRCs moved out of the communication range. The Network simulator II (NS-2) [24] is adopted to perform our simulation. We simulated MCDS as well as the method for comparison – single custodian distributed storage (SCDS) and evaluate their performance in transmitting FTP data source. In the simulation model, there is an assumption that all the custodians are with the same communication range (50m); when every two custodians are in the communication range, they can set up the connection; they stop transmitting bundles when they are out of the communication range. According to the received bundles of RC, two metrics, “goodput ratio” (GR) and “retrieved loss ratio” (RLR), were adopted to examine the efficiency of MCDS. GR is the bandwidth of the useful packets at the receiver side, which is also called the effective receiving rate. This metric is used to evaluate the QoS of MCDS. The higher GR indicates a more satisfactory QoS. RLR is the ratio of bundles failed to be retrieved during the congestion control period; this metric is used to evaluate the performance of reliability transmission of MCDS. Fig. 10 illustrated the network topology used in the simulation. The size of simulation area is $500m \times 500m$. The deployment of custodians is via normal distribution and the communication range of each custodian is within a radius of 50m. The SC uses FTP server as the data source, and the time of simulation is 300 seconds.

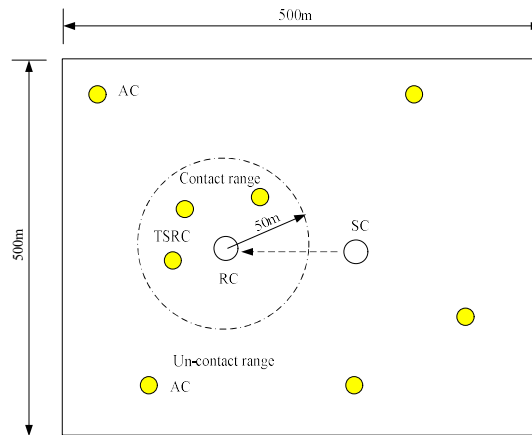


Fig. 10. The network topology

4.2 Different number of custodians

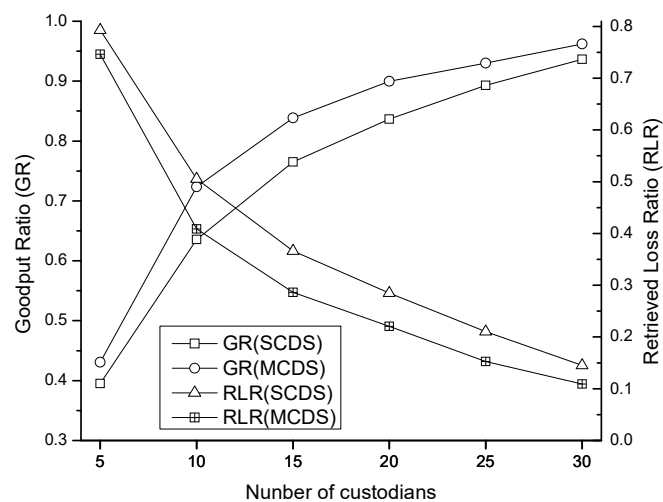


Fig. 11. Performance versus number of custodians

The number of custodians is an important factor to influence the performance of distributed storage congestion control mechanism. The operations of congestion control include removing bundles for releasing congestion status as well as retrieving migrated bundle for data reliability transmission. Basically, if there are more custodians around an RC, then the RC has a higher opportunity to migrate and retrieve bundles. The effect of MCDS caused by different number of the custodians is shown in Fig. 11. In this simulation, we set the average movement rate, the threshold ratio of reliable accessibility (the value X in Eq. 10), and the size of storage resources of each custodian as 0.05 m/sec , 0.5 , and 500 kbytes , respectively.

Fig. 11 reveals that under the condition of fewer custodians, regardless of SCDS or MCDS, the performance measures GR and RLR are worse because there is a lower chance for having custodians around the RC when it needs to migrate or retrieve bundles. When the number of custodians is increased, both GR and RLR of MCDS are improved in comparison to SCDS. Because MCDS migrates multiple duplicated bundles to TSRCs, it has a higher chance to complete the operation of bundle retrieval, and therefore the bundle loss of MCDS is lower than that of

SCDS. Under this situation, the GR of SCDS is improved by a 13.8% increment, and the RLR is improved by a 27.3% decrement. Once the amount of custodians is increased sufficiently, both MCDS and SCDS have higher opportunities to accomplish the operation of congestion control, and thus GR and RLR of SCDS are similar to those of MCDS.

4.3 Different size of storage resources

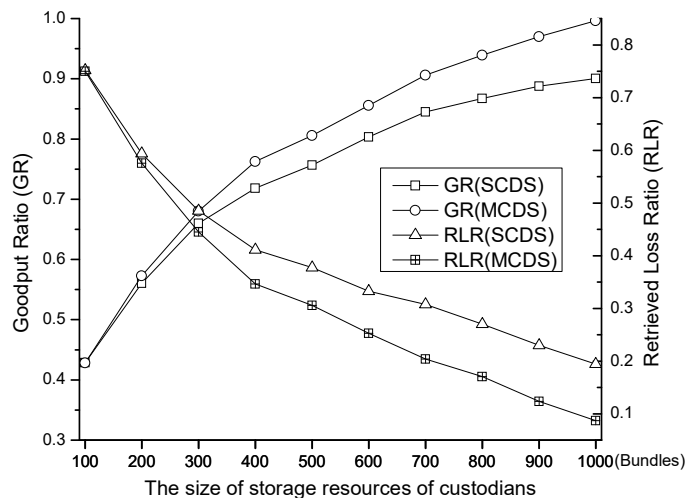


Fig. 12. Performance versus size of storage resources

The congestion DTN usually causes insufficient storage resources of RC or ACs. When an RC is congested, it needs to migrate bundles for releasing the congestion status. ACs should satisfy the requirements of having sufficient storage resources and moving in the communication range. To see the effect upon the size of storage resources of custodians, the simulation results are shown in Fig. 12. In this simulation, we set up 15 custodians in the simulation area. The average movement rate and the threshold ratios of reliable accessibility are 0.05 *m/sec* and 0.5, respectively.

According to Fig. 12, when the storage resources of custodians are small, the MCDS has worse GR and RLR and cannot improve SCDS significantly. Though the RC in MCDS has a higher chance to contact ACs than that in SCDS, it is still demanding for releasing the congestion network because these ACs do not have available storage resources to become TSRCs. Nevertheless, when the size of storage resources of custodians is increased, the performance of congestion control of MCDS is getting superior to that of SCDS. As can be seen, GR and RLR of SCDS are improved by 10.6% and 36.8% with MCDS, respectively.

4.4 Different threshold ratio of reliable accessibility

When an RC has to remove bundles for the operation of congestion control, these bundles have to be migrated to TSRCs that have high accessibilities to the RC. Because these TSRCs usually keep connecting to the RC, they are helpful for retrieving bundles when the network congestion problem is solved. In section 3.4.1, Eq. (10) is adopted to analyze the ratio of reliably accessible probability and the X in Eq. (10) is defined as the threshold ratio, a higher X value indicates that DTN has a higher opportunity to obtain reliability transmission. In this simulation, we alter the value of X to observe the affection of data transmission. We set up 15 custodians in the simulation area. The average movement rate and the size of storage resources of each custodian are set to 0.05 *m/sec* and 500 kbytes, respectively. The simulation results are shown in Fig. 13.

If the threshold ratio of reliable accessibility is a small value, the behavior of MCDS is almost the same as SCDS because MCDS has fewer TSRCs around the RC, and the simulation results also reveal that GR and RLR of the MCDS and SCDS are very similar in this situation. The simulation results of MCDS are getting better when the

threshold values of reliable accessibility are increased, since MCDS has more TSRCs for temporarily storing the bundles than SCDS (where SCDS still has one even none TSRCs around the RC). Fig. 13 shows that GR and RLR of the SCDS are improved by 11.3% and 23.2% with MCDS, respectively.

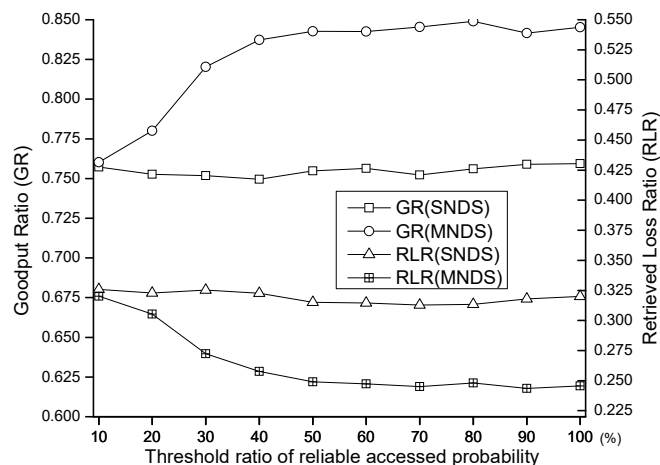


Fig. 13. Performance versus the threshold ratio of reliably accessible probability

Although a higher threshold value of reliable accessibility enables the chance of migrating or retrieving bundles in MCDS, it is not proportional to GR and RLR directly. The purpose of MCDS is to increase the accessibility when RC needs migrating or retrieving bundles. RC has to retrieve the bundle from only one of the connected TSRCs storing the migrated bundle, and thus more connected TSRCs cannot increase the number of retrieved bundles. Therefore, the threshold ratio of reliable accessibility can increase the retrieval opportunity of the migrated bundles, but is not linearly proportional to the number of the retrieved bundles. From Fig. 13, we can observe that GR and RLR of MCDS will keep steady when the threshold value of reliable accessibility exceeds 50%.

4.5 Different moving rate of custodians

The velocity of custodians changes the topology of DTN, and affects the operation of congestion control significantly. The performance of data transmission in DTN versus the velocities of the custodians is shown in Fig. 14. In this simulation, we set up 15 custodians in the simulation area; the threshold ratio of reliable accessibility and the storage resources size of each custodian are 0.5 and 500 kbytes, respectively.

Because the deployment of custodians is initialized by a normal distribution, when the velocity of the custodians is low, RC is easy to find the appropriate custodians for migrating or retrieving the bundles. Fig. 14 reveals that when the velocity of the custodians is slow, GR and RLR of MCDS are similar to those of SCDS, and both MCDS and SCDS perform good because RC can find TSRCs easily. When the custodian velocity is increased, the topology of DTN becomes unstable. Because RC is hard to find suitable TSRCs, the performance of data transmission is decreased in both methods. However, MCDS has more opportunities than SCDS to complete the operation of congestion control, and thus MCDS has a better performance.

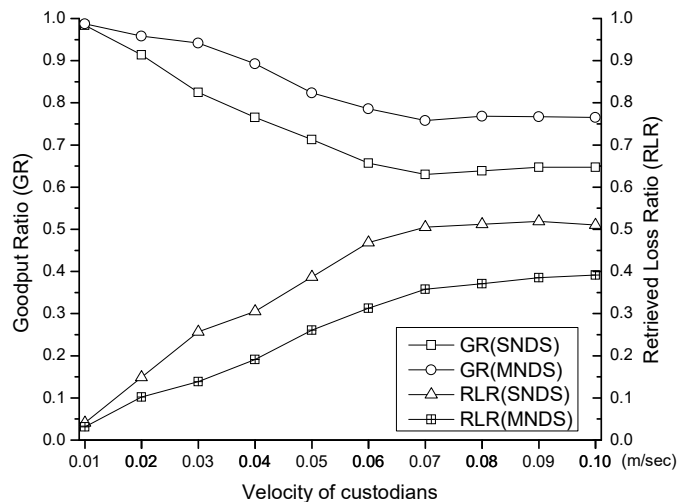


Fig. 14. Performance versus moving speed of custodians

According to the results, GR and RLR of MCDS and SCDS are getting steady when the velocity of the custodians exceeds 0.07m/sec. It is possible that when the velocity of the custodians gets higher and exceeds a specific speed, the deployment of custodians is approximate to be randomly distributed again. Hence, RC has more opportunities to find TSRCs and the simulation results are stable since the deployment of the custodians appears to be randomly distributed in the area. This phenomenon is dependent upon the environment factors such as the size of area, the number of custodians deployed and the velocity of custodians. In the steady state, The GR and RLR of the SCDS are improved by 18.4% and 27.4% with MCDS, respectively.

4.6 Summary of the performance of MCDS

From the simulation results, MCDS always outperforms SCDS. Exceptions (that they have similar GRs and RLRs) only happen in the extreme cases such as there are too few custodians in DTN, the custodian’s velocity is slow, and the storage resource is insufficient. By comparing MCDS with SCDS, the GR and RLR of SCDS are improved by MCDS consistently in different environments. The simulation results of MCDS are summarized in Table 1.

Table 1. Summary of simulation results of MCDS to improve SCDS in different environments

	Number of custodians	Storage resources size of custodian	Reliable accessibility	Moving rate of custodians
GR	13.8%	10.6%	11.3%	18.4%
RLR	27.3%	36.8%	23.2%	27.4%

5.0 CONCLUSIONS

In this paper, MCDS approach is proposed as a new congestion control mechanism of DTN. MCDS uses the storage resources provided by the neighbor nodes and spends a little extra transmission time to solve DTN congestion problem and improve the transmission reliability simultaneously. MCDS selects multiple custodians for storing the duplication of migrated bundles. These duplicated bundles increase the probability for successfully retrieving the required bundles when the congestion status is released. The simulation results reveal that MCDS is effective for congestion control and enhances the reliability of transmission in DTN. Nowadays, some networks are extensions of DTN and these networks are suitable to adopt MCDS. Examples include Deep Space Networks (DSN), Vehicular Ad hoc Networks (VANET), and Under Water Acoustic Sensor Networks (UW-ASN). However, the issue of congestion control still remains in these special DTNs. Following the successful application of the principle of

MCDS to DTN, we plan to design the congestion control mechanisms for the above DTN extensions in the future.

REFERENCES

- [1] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay-Tolerant Networking Architecture", *IETF RFC 4838*, 2007.
- [2] K. Scott and S. Burleigh, "Bundle Protocol Specification", *IETF RFC 5050*, 2007.
- [3] E. Coe and C. Raghavendra, "Token Based Congestion Control for DTNs", *Proceedings of 2010 IEEE Aerospace Conference*, 2010.
- [4] A. G. Voyiatzis, "A Survey of Delay- and Disruption-Tolerant Networking Applications", *Journal of Internet Engineering*, Vol. 5, No. 1, June 2012, pp. 331-344.
- [5] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss, "Delay-tolerant networking: an approach to interplanetary Internet," *IEEE Communications Magazine*, vol. 41, no. 6, 2003, pp. 128–136.
- [6] R. Wang, Z. Wei, Q. Zhang and J. Hou, "LTP Aggregation of DTN Bundles in Space Communications", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 49, Issue 3, 2013, pp. 1677 – 1691.
- [7] F.J.L. Ribeiro, A.C.P. Pedroza and L.H.M.K. Costa, "Deepwater Monitoring System in Underwater Delay/Disruption Tolerant Network", *Latin America Transactions, IEEE (Revista IEEE America Latina)*, Vol. 10, Issue 1, 2012, pp. 1324 – 1331.
- [8] M.J. Khabbaz, H.M.K. Alazemi and C.M. Assi, "Delay-Aware Data Delivery in Vehicular Intermittently Connected Networks", *IEEE Transactions on Communications*, Vol. 61, Issue 3, 2013, pp. 1134 – 1143.
- [9] P. Juang, H. Oki, and Y. W. et al., "Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet", *Proceedings of the 10th international conference on Architectural support for programming languages and operating systems*, Vol. 36, Issue 5, December 2002, pp. 96-107.
- [10] L. Yun, C. Xinjian, L. Qilie and Y. Xiaohu, "A Novel Congestion Control Strategy in Delay Tolerant Networks", *Proceedings of Second International Conference on Future Networks, ICFN '10*, 2010, pp. 233-237.
- [11] M. Seligman, K. Fall, and P. Mundur, "Storage routing for DTN congestion control", *Wireless Communications & Mobile Computing - Wireless Ad Hoc and Sensor Networks archive*, Vol. 7, Issue 10, December 2007, pp. 1183-1196.
- [12] D. Hua, X. Du, G. Xu, L. Cao and H. Chen, "A DTN congestion mechanism based on Distributed Storage", *Proceedings of the 2nd IEEE International Conference on Information Management and Engineering, ICIME 2010*, 2010, pp. 385-389.
- [13] B. Soelistijanto and M.P. Howarth, "Transfer Reliability and Congestion Control Strategies in Opportunistic Networks: A Survey", *Communications Surveys & Tutorials, IEEE*, Vol. 16, Issue 1, 2014, pp. 538-555.
- [14] M. Allman, V. Paxson, and E. Blanton, "TCP Congestion Control", *IETF RFC 5681*, 2009.
- [15] P. Natarajan, F. Baker, P. D. Amer, and J. T. Leighton, "SCTP: What, Why, and How", *IEEE Internet Computing*, vol.13, no.5, 2009, pp.81-85.
- [16] Y.-C. Lai, "DCCP: Transport Protocol with Congestion Control and Unreliability", *IEEE Internet Computing*, vol.12, no.5, pp.78-83, 2008.

- [17] M. Seligman, K. Fall, and P. Mundur, "Alternative custodians for congestion control in delay tolerant networks", *Proceedings of the 2006 SIGCOMM workshop on Challenged networks, CHANTS '06*, 2006, pp. 229-236.
- [18] K. Fall, "A Delay Tolerant Network Architecture for Challenged Internets", *Proceedings of ACM SIGCOMM*, August 2003, pp. 27-34.
- [19] I. Bisio, M. Cello, T. de Cola, and M. Marchese, "Combined Congestion Control and Link Selection Strategies for Delay Tolerant Interplanetary Networks", *Proceedings of the 28th IEEE conference on Global telecommunications, GLOBECOM'09*, 2009, pp. 2384-2389.
- [20] Q. Yang, F. Qin, J. Bi and Q. Xu, "Adaptive Parameter Estimation Based Congestion Avoidance Strategy for DTN", *Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering, ICCSEE 2013*, January 2013, pp. 200-205.
- [21] J. Ott and M. J. Pitkanen, "DTN-based Content Storage and Retrieval", *Proceedings of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, June 2007.
- [22] Y. Yang, L. Han, W. Xu, X. Kong and R. yan, "An Advanced Congestion Control Mechanism Based on Distributed Storage for DTN", *Proceedings of the 2012 International Conference on Computer Science and Service System, CSSS '12*, 2012, pp. 1285-1289.
- [23] L. Yin, H. Lu, Y. Cao and J. Gao, "An Incentive Congestion Control Strategy for DTNs with Mal-behaving Nodes", *Proceedings of Second International Conference on Networks Security Wireless Communications and Trusted Computing, NSWCTC 2010*, 2010, pp. 91-94.
- [24] The Network Simulator– ns-2,
<http://www.isi.edu/nsnam/ns>, 2015