Route Caching in DTNs Interconnected by Infrastructure

Md. Enamul Haque[1], Yasser Kamal Hassan[2], Anika Aziz[3], Cristian Borcea[4] and Shigeki Yamada[5]
[1]Bangladesh Agricultural University, Mymensingh, Bangladesh
Email: enamul@bau.edu.bd
[2]South Valley University, Qena, Egypt
Email: yasser.moawad@sci.svu.edu.eg
[3]University of Dhaka, Dhaka, Bangladesh
Email: anika.aziz@gmail.com
[4]New Jersey Institute of Technology, New Jersey, USA
Email: borcea@cs.njit.edu
[5]National Institute of Informatics, Tokyo, Japan
Email:shigeki@nii.ac.jp

***Abstract*— Numbers of mobile devices are sharply increasing in recent years. To deal with this mobility tendency, Delay and Disruption Tolerant Networks (DTN) has received significant research focus. In that context, infrastructure based DTN has promising application perspectives. In this paper, we present a route caching mechanism for infrastructure based DTN communication where the infrastructure connects multiple DTN regions. Our proposed mechanism reduces flooding of DTN messages in the infrastructure significantly when message are passed through the infrastructure. We evaluated this route caching mechanism in three flooding based DTN routing protocols. Simulations on an actual urban map show that the delivery ratio of the whole network increased greatly (by 25% to 81%) as well as overhead also decreases (maximum 50%).**

Keywords*— *infrasrtucture; route caching; flooding

I. INTRODUCTION

Advancement in the field of computing and communication has greatly expanded the use of wireless mobile devices. Cisco has forecasted that there will be 7.1 billion mobile devices in 2015 which will be equal to the world's population [1]. These mobile devices include laptops, smartphones, tablets, gaming consoles. Such trend highlights the need for flexible and efficient mobility services in the future Internet. Mobility induces fluctuations in connectivity and may lead to complete disconnection. To mitigate the mobility challenge, enhancements and modifications with the existing TCP/IP has been proposed and tested. However, the proposed techniques are yet to be proven as practically efficient. In this regard, Delay and Disruption Tolerant Networks (DTN) takes a different approach of relying exclusively on asynchronous communication. DTN communication model is based on two advanced capabilities, namely custody transfer and hop by hop routing. Custody transfer allows messages to be buffered in DTN nodes until forwarded or found to be unnecessary. In hop by hop routing, routing decisions are made dynamically at each hop. These unique features of DTN are quite promising to deal with mobility of the future Internet. Many research projects are currently working on future Internet issues considering DTN for managing challenged network conditions [2].

For wider applicability and performance improvement of DTN, there have been efforts from diverse perspectives such as use infrastructure elements. Infrastructure in DTN context refers to fixed nodes with plenty of resources and connected to other fixed nodes through dedicated links. This infrastructure network can be connected to the Internet or some other WAN and can be accessed wirelessly via WiFi or cellular networks. Such access to the infrastructure increases contact opportunities in DTN as well as infrastructure traffic can be offloaded too. Moreover, an infrastructure can connect DTN regions located in a sparse area in which they cannot communicate to each other directly. As a good number of DTN protocols use flooding based mechanism, connecting DTN with infrastructure may cause unnecessary spreading of messages which may cause traffic overflow. To keep the traffic stable in the infrastructure, we propose a route caching mechanism for the infrastructure which connects multiple DTN regions. Our proposed route caching mechanism caches route information in the fixed nodes of the infrastructure and utilizes that information when messages are forwarded using same route consecutively. This simple mechanism is found to be effective in simulation based evaluations. The rest of the paper is organized as follows: Section 2 describes existing proposition of infrastructure elements in DTN, Section 3 presents the problem statement, Section 4 mentions the related work, Section 5 describes our considered communication model, Section 6 describes the proposed route caching algorithm, Section 7 presents the evaluation of our proposition and Section 8 concludes the paper with future directions.

II. RELATED WORKS

The route caching mechanism is an existing mechanism in the IP network [17] which is regaining its importance due to sweeping growth of routing tables in the Internet routers. According to the caching strategy, frequently used routes and other routes are stored in larger but slower memory and utilized in forwarding. The Internet requires very fast packet forwarding rate, thus this route caching strategy is in challenging situation. However, the application of our route caching mechanism in DTN has different objectives. Due to the nature of DTN, very fast packet forwarding is not mandatory. Rather providing connectivity to directly inaccessible DTN regions with better delivery ratio is more important. Caching strategy has also been utilized in Cellular IP [18]. Cellular IP has been proposed to deal with node micro mobility and uses two types of caches in the base stations, namely page cache and route cache. Whenever a mobile node moves from one paging area or cell to another paging area or cell, updates are sent to gateway node(s) through base stations and their caches are updated with node location and/or node state. Thus, these caches are used for the location management at a micro level. Our proposition does not deal with node location at this stage, rather connects regions (much bigger than cells in cellular IP) eliminating unnecessary spreading of messages. There is also some resemblance with the AODV [19] protocol's route table management. AODV creates end-to-end path from source to destination through route discovery phase. However, proposed route caching routes messages in a hop by hop manner and employs comparatively simpler technique.

DTNs connected to the fixed infrastructure are proved to be beneficial [15]. There are several ways to utilize the infrastructure. In [6][7], stationary rendezvous nodes have been used as information sink which are located in wide geographical area. These nodes connect mobile devices to the network through strong radio quality signal. A similar approach has been proposed in [10][11] where stationary nodes, termed as throwboxes, are placed strategically - not connected with each other and act as message relays following the store and forwarding paradigm. Another technique of using the infrastructure is the use of message ferries with fixed nodes [8][9]. These message ferries are infrastructure elements and relay messages between mobile nodes or mobile node to fixed nodes. Furthermore, a framework for comparison of infrastructure elements has presented in [12] where three infrastructure enrichments were compared. The analysis from this comparison showed that end-to-end delay can be reduced by a factor of 2 through addition of x base stations, the same reduction required 2x mesh nodes or 5x throwboxes. Thus, use of infrastructure in DTN is considered beneficial and a well motivated research issue. In our route caching proposition, we are not using any supporting elements like message ferries, throwboxes. Rather, we employ route caching to restrict flooding of messages throughout the infrastructure and utilize the existing infrastructure in connecting DTN regions in a DTN way (hop by hop routing).

We have utilized this caching mechanism for a new purpose in DTN routing. It makes the exiting DTN routing solutions not directly comparable to our proposition. Rather existing DTN routing protocols, more specifically flooding based protocols, can be benefited from the proposed route caching mechanism when fixed infrastructure is included in the network scenario. Epidemic routing protocol is solely based on the information exchanges between two encountering mobile nodes and thus distributing messages throughout the network to reach the destination [3]. There is no limitation about the number of copies a message in the network. The Spray and Wait (SnW) protocol adds limited copy flooding feature to the mobile nodes while routing to the destination [4]. The Spray and Focus scheme distributes even a small number of copies to few relays [5]. Each relay can then forward its copy further using a single-copy utility-based scheme, instead of naively waiting to deliver it to the destination by itself as happened in [4]. These flooding based routing protocols make use of only localized knowledge and hence suffer from reduced delivery ratio and large latencies. Handoff-based And Limited Flooding (HALF) protocol [14] uses the general handoff mechanism intended for the IP network, in a DTN way and also integrates a limited flooding technique to it. It utilizes some knowledge on destination nodes (handoff mechanism) and some knowledge on topological information though use of cache tables. However, tables on HALF are only updated on edge routers of infrastructure where handoff takes place. And, if HALF has no knowledge about route, it uses limited flooding. Thus, above mentioned flooding based protocols rather can utilize the proposed route caching mechanism and attain better performance.

III. Problem statement

DTN routing protocols rely on flooding based mechanism or us probabilistic estimations to deal with uncertain contact opportunities. Flooding mechanism spread many copies of a message across the network and increases the chance to deliver that message to the destination significantly. Among such DTN protocols, Epidemic protocol [3] copies messages in each hop whereas Spray and Wait [4], HALF [14] use limited number of messages by controlled flooding. Though flooding in the fixed networks is strictly avoided, in case of DTN, it is proved to be effective [13]. When DTN is combined with a fixed infrastructure, any sort of flooding of messages expectedly creates high volume of traffic and can overflow the infrastructure network. Unlike DTN regions, fixed infrastructure has dedicated fixed links. So, the network topology is pretty much fixed which makes flooding unnecessary and unrealistic. To mitigate this problem, we present a route caching mechanism for the DTN-fixed infrastructure communication.

IV. Our communication model

In our communication model, multiple DTN regions are located in geographically distanced positions. Mobile nodes in each region can communicate to each other based on opportunistic contact opportunities. However, communication from one region to another region is only possible through the

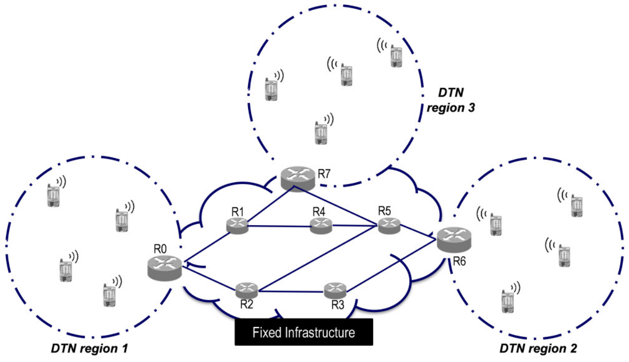

Fig. 1. DTN regions connected via fixed infrastructure

fixed infrastructure. When a DTN message travels through the infrastructure, it traverses in hop by hop manner like inside DTN regions. As DTN protocols tend to produce multiple copies of a message, proposed route caching mechanism caches route information from message traversals and maintains a cache table in each node in the infrastructure. This cache table is not shared with neighbor nodes or any other node. Fig. 1 depicts such communication model where three DTN regions are located far apart and cannot communicate to each other directly. With support from fixed infrastructure, nodes in DTN region 1 can communicate with DTN region 2 or 3 and vice versa. One or more fixed nodes of the infrastructure can be accessible from the each DTN regions. The fixed infrastructure can use a different routing protocol which is more suitable for fixed connections. While to deal with DTN messages, the proposed route caching mechanism is utilized in fixed nodes (routers) in infrastructure. In the presence of route caching mechanism, DTN messages are not unnecessarily flooded throughout the infrastructure. For example, if a message is created in DTN region 1 and destined to a node in DTN region 2 in Fig. 1, it first reaches to the fixed router R0 through opportunistic contact opportunities. If R0 has any information in the cache table about the destination and related next hop, the message is forwarded to the next hop. If there is no information, the message is flooded. In case of Spray And Wait or HALF protocol, the message is flooded with limited number of copies. This number of copies is decreased in each hop in binary mode [4] and forwarded till the corresponding node has a single copy. In every hop, a message source, destination and next hop id is stored in cache table. Now, if any message has traversed from this destination to the source, this path in cache table is marked as ok. Thus, for consecutive messages from same source in DTN region 1 to same destination in DTN region 2, messages are forwarded based on cache table inside the infrastructure network.

V. THE ROUTE CACHING BASED ALGORITHM

The route caching algorithm acts in each fixed node of the infrastructure. This algorithm has two parts, Algorithm 1 and 2 as depicted in Fig. 2 and 3. Algorithm 2 is called from the Algorithm 1. As depicted in Algorithm 1, upon receiving a message M, the node tries to forward to every connection. If a connection is established and M is forwarded via that connection, M's source ID, destination ID is cached as a record (entry) in the cache table. Moreover, that connection's other end node ID is also stored as next hop in the same record. Additionally, that record's flag is set as NULL as this route is not cross checked by a returning message. Therefore, this caches information is not usable yet. Step 12 to 18 of Algorithm 1 show how route information is cached when a message is traversed through this concerning node for the first time. For every message M, the corresponding node also checks whether M's destination ID and any cached record's considered as a returning message. In this case, cached record's next hop is set with M's previous hop ID and record's flag is set as OK. Step 5 to 7 shows this cache table update process when destination node is sending a returning message to the source node. Typically, such returning message can be an acknowledgment message.

```
 1: procedure RouteCaching
 2:     Look up $cachetable$
 3:     while Receiving message $M$ do
 4:         if $M.dst = cachedRecord.dst$ AND $cachedRecord.flag = OK$ then
 5:             $M.nextHop \leftarrow cachedRecord.nextHop$
 6:             if $M.dst = cachedRecord.source$ then
 7:                 $cachedRecord.nextHop \leftarrow M.previousHop$
 8:                 $cachedRecord.flag \leftarrow OK$
 9:                 for every message $M$ do
10:                     if $M.nextHop = TRUE$ then
11:                         Do $Algorithm2$
12:                         Forward $M$ to nextHop
13:                     else
14:                         Try $M$ for every connection
15:                         if Connection $conn$ is available then
16:                             Create new $cachedRecord$ in cache table
17:                             $cachedRecord.src \leftarrow M.src$
18:                             $cachedRecord.dst \leftarrow M.dst$
19:                             $cachedRecord.nextHop \leftarrow conn.otherEndNode$
20:                             $cachedRecord.flag \leftarrow NULL$
21:                             Do $Algorithm2$
22:                             Forward $M$ to $conn.otherEndNode$
23:                         end if
24:                     end if
25:                 end for
26:             end if
27:         end if
28:     end while
29: end procedure
```

Fig. 2. Route caching algorithm

```
 1: procedure Forwarding
 2:     Look up cache table
 3:     if protocol: $SprayandWait$ OR $HALF$ then
 4:         if corresponding node is mobile node then
 5:             $numberofcopiesofamessage \leftarrow numberofcopies/2$
 6:             forward $M$ with number of copies
 7:         else
 8:             forward $M$ with number of copies
 9:         end if
10:     else
11:         forward $M$
12:     end if
13: end procedure
```

Fig. 3. Forwarding algorithm

Upon receiving every message the corresponding node also looks up the cache table to find a match of M's destination ID and a cached record having same destination ID. If there is a match, this message is considered as a consecutive message

TABLE I. SIMULATIONS PARAMETERS

Parameters	Values
Message size	500KB - 1MB
Message generation interval	1-6 min.
Message lifetime	12 hours
Message generation duration	Till 15th hour
Transmission speed	2 Mbps (mobile nodes) 100 Mbps (fixed routers)
Number of mobile nodes (each region)	5
Number of fixed routers	15
Node speed	0.5 – 15 m/sec.
Buffer capacity	200MB (mobile nodes) 2GB (Fixed routers)
Waiting time slots	15 – 30 min. 6 – 8 hours
Wireless range	100m
Simulation duration (real time)	24 hours
Warm up times	3 hours

(2nd or 3rd or so on). If such match is found and corresponding record's flag is marked as OK, messages M's next hop field is set with the corresponding record's next hop ID. This is possible because previous traversal of message is cached in the cache table as mentioned previously. This cache table based forwarding of M is described in step 2 to 4 in Algorithm 1.While a message M is forwarded, the node follows Algorithm 2. SnW and HALF routing protocol use limited flooding by fixing initial number of copies of M. Whenever M is forwarded, this number of copies is divided by 2 - receiver gets half and sender keeps half. For example, node A is sending message M7 to node B. node A has 8 copies of M7. During the transmission, node A keeps 4 copies and node B gets 4 infrastructure (fixed nodes). The motivation is: fixed node uses route caching. There is no need to decrease the number of copies. So, according to Algorithm 2, numbers of copies are decreased if next node is a mobile node (depicted in step 2 to step 4 in Algorithm). Otherwise, M is forwarded with number of copies value same.

VI. PERFORMANCE EVALUATION

A. Simulation configurations

To evaluate the performance of the proposed route caching mechanism, we performed extensive simulations in ONE simulator [16]. We constructed simulation models for route caching algorithm in Epidemic, SnW and HALF protocols and compared performance with and without route caching mechanism. An actual portion of Tokyo city map was used to simulate fixed infrastructure and mobile nodes in a city. DTN region 1 and 2 were created (marked in Fig. 4) each having 5 mobile nodes. The fixed infrastructure had 15 fixed routers connected through 21 dedicated links. Message flow was from one DTN region to another DTN region through the fixed infrastructure and vice versa. Messages with random size were assumed to be pedestrians, cars, buses. So, realistically the movement speed was from 0.5 to 15 meter per seconds. Every mobile node randomly chose a destination point on the map from a list of points and moved towards that point (black squares in Fig. 4). These points are generally termed as points of interest (POI) and represents frequent meeting points like stations, shopping malls, banks in a city. Fixed routers of infrastructure were placed near these POIs. Once a node reached such intermediate destination point, it waited for certain period of time and then chose another destination point. To correspond to the real world nodes' waiting patterns, this waiting time was set from two slots such as 15 to 30 minutes and 6 to 8 hours. Any one slot of waiting time was randomly selected and certain duration was also randomly set from the range of that slot. Total simulation duration was 24 hours with cache table in fixed routers refresh interval 1 hour. Further details of simulation configuration are given in Table 1.

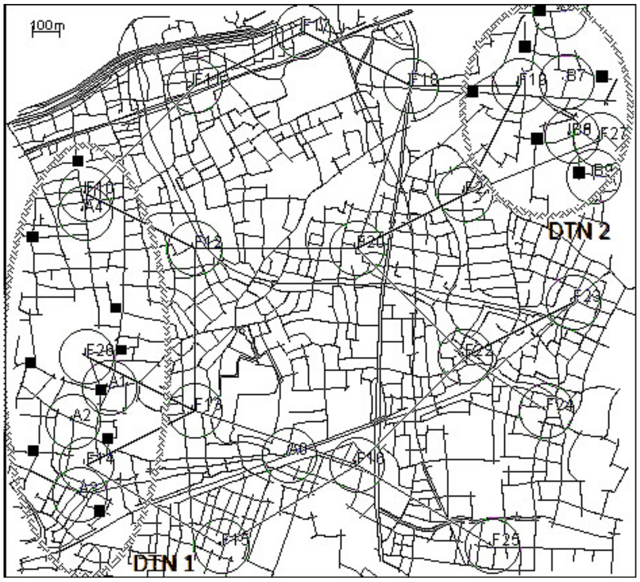

Fig. 4. Simulation scenario (part of Tokyo)

Before running actual simulations, we conducted some test runs to identify when message traffic reached a stable state. Because gathering simulation statistic from very beginning without reaching a stable traffic may produce biased and incomplete results. Our prior test run results showed that overhead ratio reached a stable traffic state after the 3rd hour. So, simulation statistic was collected after 3rd hour to end of simulation. In setting the initial number of copies of a message in SnW and HALF protocol, a single message was stored in the buffer and number of its copy was set as a message property. For these two protocols, this value was set as 30. To eliminate further statistical biasness, each simulation was run with three different movement model seeds. Then, those three run results were averaged to prepare our final evaluation results.

B. Results

When basic Epidemic protocol was run in all nodes including mobile nodes and fixed nodes in infrastructure, the

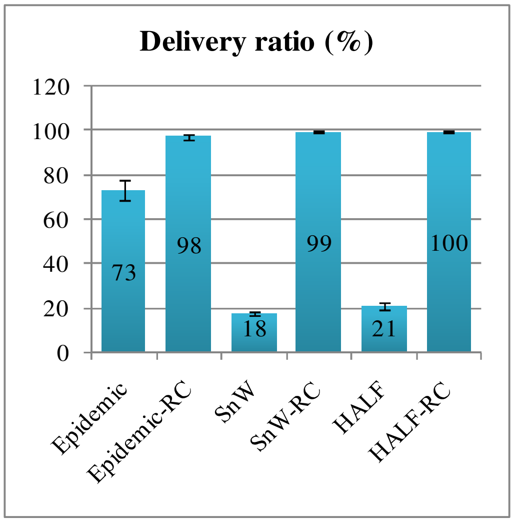

Fig. 5. Delivery ratio of three protocols

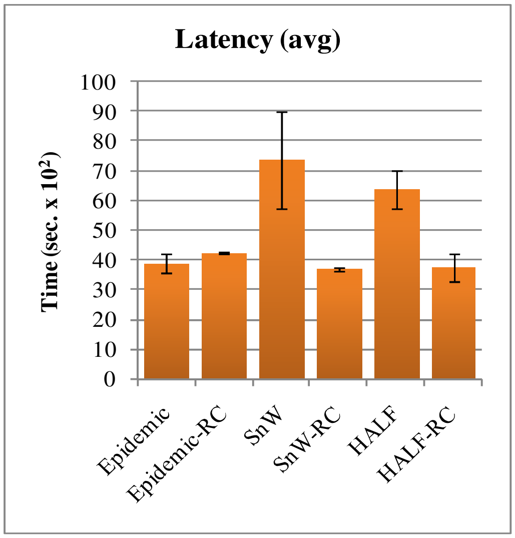

Fig. 6. Average latency of three protocols

delivery ratio was 73%. However, with route caching mechanism, the delivery ratio reached 98%. SnW protocol and HALF protocol had number of delivered message 87 and 100 respectively which is 18% and 21% respectively without route caching (RC). On the contrary, the number of delivered message was 479 (99%) and 480 (100%) with route caching as shown in Fig. 5. The average latency dropped from 7365 seconds to 3707 seconds and from 6366 seconds to 3742 seonds by incorporating RC mechanism respectively in SnW and HALF. However, this latency slightly increased in Epidemic which was from 3881 seconds to 4234 seconds as shown in Fig.6. The overhead ratio was calculated by (1.0*(number of forwarded message delivered/delivered)). This measure reflects overheard in relation to delivered message. SnW and HALF protocol had this overhead between 70 to 80 without route caching. With utilization of route caching protocol, this ratio dropped to near 25. In case of Epidemic protocol, this difference of overhead ratio between having and not having route caching was around 74 as shown in Fig. 7. In the Fig. 5, 6, 7, error bars show deviations of simulation results of multiple runs from the average values (pick of the bars). In most cases, these deviations were quite minimal. The amounts of traffic through infrastructure and cache table hit rate were also measured. The cache table hit ratio among infrastructure traffic were 6.2%, 6.5% and 7.3% in Epidemic, SnW and HALF protocol respectively with route caching (shown in Fig. 8).

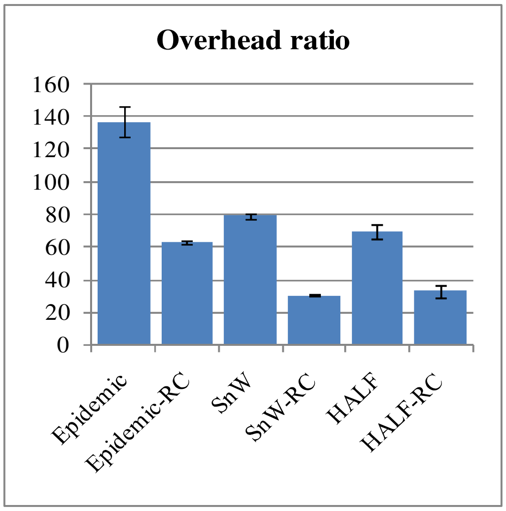

Fig. 7. Overhead ratio of three protocols

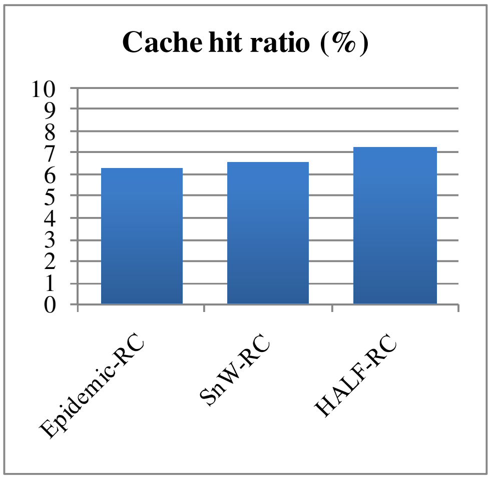

Fig. 8. Cache table hit ratio of three protocols

C. Analysis

The simulation results show that protocols utilized with RC mechanism performed significantly well in terms of delivery ratio and latency. In SnW and HALF, route caching boosted performance by 81% and 79% respectively compared to base protocols. This difference was 25% in Epidemic. With RC, messages were tended to be forwarded towards proper destination. As a result, more messages reached destination before expiration and greatly contributed on better delivery ratio. Route caching mechanism greatly affected another vital performance metric that is latency. The latency was reduced by 50% and 41% respectively in SnW and HALF. However, there was a marginal increase of latency (8%) in case of Epidemic protocol. Another important performance metric is overhead ratio. In all three protocols, RC mechanism significantly reduced overhead ratio as shown in Fig. 7. In case of Epidemic, this caching mechanism proved to be very effective by dropping overhead ratio from 137 to 63. The cache hit ratio was measured as the ratio between numbers of messages forwarded based on cache table and numbers of messages forwarded in total - through the infrastructure in both cases. This metric indicates how effectively cache tables were utilized while routing messages. HALF has the highest cache table hit ratio (7.3%) and proved to be most effective with RC mechanism. Other two protocols had cache table hit ratio over 6% as shown in Fig. 8. In practical applications, the processor hit ratio is much higher even close to 100%. The cache hit ratio of the proposed mechanism seems rather low. One reason is that the cache table hit ratio greatly depends on the message traffic pattern. Secondly, a route was only used if it had been confirmed with a return message from corresponding destination to the source. Considering these performance measures, it is clearly evident that the route caching mechanism significantly improves routing performance in infrastructure based DTN communication.

VII. Conclusion

In this paper, we have presented a route caching mechanism that can be utilized in an infrastructure network which connects multiple DTN regions. This caching mechanism works very effectively in communication between DTN regions when flooding based protocols are used. Three flooding based protocols were incorporated with this mechanism and their performances ware analyzed. The delivery ratio with the proposed mechanism was quite impressive and it also reduced latency and delivery overhead. As our next step, we shall analyze our proposition considering different performance factors such as congestion, more DTN regions, and wider infrastructure area.

References

[1] Cisco Systems Inc. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2010-2015, http://newsroom.cisco.com/dlls/ekits/Cisco_VNI_Global_Mobile_Data_Traffic_Forecast_2010_2015.pdf, February 2011.

[2] NSF's FIA: http://www.nets-fia.net, March 2013.

[3] A. Vahdat and D. Becker, "Epidemic routing for partially connected ad hoc networks," Technical Report CS-200006, Duke University, April 2000.

[4] T. Spyropoulos, K. Psounis, and C. S.Raghavendra, "Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks," ACM SIGCOMM WDTN, pp. 252-259, August 2005.

[5] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and Focus: Efficient Mobility-Assisted Routing for Heterogeneous and Correlated Mobility," IEEE PerComW, pp. 79-85, March 2007.

[6] T. Small and Z.J. Hass, "The Shares Wireless Infostation Model-A new Ad-hoc networking paradigm (or While there is a Whale, there is a way)," ACM MobiHoc'03, pp.233-244, June 2003.

[7] D. Goodman, J. Borras, N. Mandayam, and R. Yates, "INFOSTATIONS: A New system model for data and messaging services," IEEE VTC'97, vol.2, pp. 969-973, May 1997.

[8] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," ACM MobiHoc'04, pp.187-198, October 2004.

[9] F. Yasmeen, N. Huda, C. Borcea, and S. Yamada, "Ferry access points and sticky transfers: Improving communication in ferry-assisted DTNs," IEEE WoWMoM AOC'12, pp.1-7, June 2012.

[10] N. Banerjee, M. D. Corner, and B. N. Levine. "An Energy-Efficient Architecture for DTN Throwboxes," IEEE INFOCOM, pp.776-784, May 2007.

[11] W. Zhao, Y. Chen, M. Ammar, M. Corner, B. Levine, and E. Zegura, "Capacity Enhancement using Throwboxes in DTNs," IEEE MASS, pp. 31-40, October 2006.

[12] N. Banerjee, M. D. Corner, D. Towsley, and B. N. Levine, "Relays, Base Stations, and Meshes: Enhancing Mobile Networks with Infrastructure," ACM MobiCom'08, pp.81-91, September 2008.

[13] E. P. C. Jones and P. A. S. Ward, "Routing strategies for delay-tolerant networks," ACM SIGCOMM, 2004.

[14] A. Aziz, M. E. Haque, C. Borcea, Y. K. Hassan, and S. Yamada, "Managing disconnected mobile nodes in a Delay Tolerant Network with HALF routing protocol," IEICE Trans. on Communications, E96-B(7), p.1691-1705, July 2013.

[15] C. P. Mayer, Hybrid Routing in Delay Tolerant Networks, KIT Scientific Publishing, July 2012.

[16] A. KerÃ¤nen, J. Ott, and T. KÃ¤rkkÃ¤inen, "The ONE Simulator for DTN Protocol Evaluation," SIMUTools'09, March 2009.

[17] C. Kim, M. Caesar, A. Gerber, and J. Rexford, "Revisiting Route Caching: The World Should Be Flat," 10th International Conference on Passive and Active Network Measurement, pp.3-12, 2009.

[18] A. G. Valko, "Cellular IP - A New Approach to Internet Host Mobility," ACM Computer Communication Review, January 1999.

[19] Perkins, E. B. Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", IETF. RFC 3561. Retrieved 2010-06-18.

.