DT-ICAN: A Disruption-Tolerant Information-Centric Ad-Hoc Network

Yu-Ting Yu*†, Joshua Joy*, Ruolin Fan*, You Lu*, Mario Gerla*, M. Y. Sanadidi*
*Computer Science Department
University of California
Los Angeles, CA
†Qualcomm Research
Bridgewater, NJ
Email: {yutingyu,jjoy,ruolinfan,youlu,gerla,medy}@cs.ucla.edu

***Abstract*—Recently, Information-Centric Network (ICN) has been attracting much attention with its promising future as next-generation Internet architecture. While ICN is scalable and efficient in Internet, concerns are raised when ICN is deployed in frequently-disruptive vehicular ad-hoc networks. In this paper, we introduce DT-ICAN, which provides low-cost, bandwidth-efficient network operations to conquer disruptions while preserving the context awareness of ICN. We implement DT-ICANSIM, an open-source simulator for DT-ICAN in QualNet. We evaluate the trade-off between the performance gain in data availability and the overhead due to epidemic dissemination using real-world traces. Our results show that DT-ICAN improves the file retrieval rate by 45% compared to traditional multi-hop ICN in real-world scenario, proving the necessity of providing an opportunistic networking option when end-to-end connectivity is unavailable for VANET ICN.**

I. INTRODUCTION

Information Centric Networking (ICN)[1] is a recently emerging research field. In ICN, a data object is retrieved based on its content identity instead of the IP address of the node on which it resides. While prior research has worked on Internet based ICN applications, the mobile ICN technology has received increasing attention due to the growing importance of mobile commercial and tactical applications. The ICN could potentially benefit mobile, connectivity challenged applications by providing in-network caching and flexible multi-source content downloading. [2][3][4][5].

In this paper, we apply the ICN technology for Vehicular Ad-Hoc Networks (VANET). Many VANET target applications (e.g. location-based crowdsourcing, urban surveillance, etc) are information-centric. Moreover, they require application and context awareness for best performance. While a vast body of existing ICN MANET studies [2][6][7] have attempted to adapt the conventional end-to-end on-demand data retrieval model to MANETs by modifying the multi-hop protocol, few have considered and evaluated the case when the connectivity between data consumer and data producers is truly disruptive. The major challenge in VANET is the potentially persistent network partitioning due to the traffic and obstacles between communication pairs. In such situation, the end-to-end path between the data consumer and any data holder cannot be established. This issue is especially severe in sparse scenarios, for example, during non-rush hour and in rural area. The solution is to rely on opportunistic dissemination to retrieve the contents.

The current ICN architecture relies on continuously retransmitted interests to retrieve data under intermittency to ensure robust delivery and to recover from network partitioning. However, when end-to-end path is unavailable, the frequent retransmissions produced by ICN's chunk-by-chunk pulls may significantly degrade ICNs scalability and efficiency. Furthermore, at any given time, the propagated data chunks are exact matches of the sent interests. Therefore, even if multiple paths could be used, all relays on these paths may only pull and cache the same data chunks, leading to lower cached chunk diversity and utilization. In this paper, we propose an all-broadcast ICN network architecture, Delay-Tolerant Information-Centric Ad-Hoc Network (DT-ICAN), which allows opportunistic networking to resolve the disruptions by per-node data-object-level interest summary and hop-by-hop randomized data chunk delivery. DT-ICAN uses the hierarchical naming of ICN and consequently preserves its context awareness: by examining the name prefixes, the network has the power to judge the application types and associates the data to the application's properties, which can be further used to optimize the network performance.

The contribution of this paper is three-folds. First, we propose a disruption-tolerant ICN architecture with hierarchical naming, DT-ICAN, which leverages the node-based interest aggregation and epidemic interest dissemination to overcome network partitions in VANET ICN. Second, we make available the DT-ICAN implementation in QualNet 6.1. The simulator, DT-ICANSIM, is publicly available on Github (https://github.com/uclanrl/dt-icansim). Finally, we evaluate and compare the performance of epidemic DT-ICAN versus end-to-end, multi-hop routing based ICN using real San Francisco taxi traces [8] as well as synthetic car traces in a Washington DC map created by VanetMobisim. This is to our knowledge the first multi-hop versus epidemic VANET ICN comparison. Our results show that epidemic-style communication is necessary for VANET ICN, and suggest to alternate between epidemic and conventional multi-hop approach adaptively.

The organization of this paper is as follows. We discuss the related work in II. In section III, we give an overview of the system architecture and discuss the major design component implementation. In section IV, we present the evaluation results. Finally, we conclude in section V.

II. Background

A. ICN Background

ICN, also known as CCN (Content Centric Network) and NDN (Named Data Network) [1], is a "new slate" network layer design aiming at replacing the current TCP/IP architecture. ICN uses data name directly instead of host address to find and retrieve data. It strictly assumes a pull-based, chunk-by-chunk transport. Every data chunk has a unique name. A requester must send an Interest packet, which carries the data chunk name, to retrieve the corresponding data chunk. Data chunks are cached along the way by all relays when they come back to requester following the interest "breadcrumbs".

All ICN nodes include three data structures: Content Store (CS), Pending Interest Table (PIT), and Forwarding Information Base (FIB). CS is the cache used to store received data. With distributed, replicated caches, server failure or intermittent disconnection can be quickly recovered. Moreover, popular contents are generously replicated among several caches mitigating server load. PIT "remembers" all pending interests a node received. The two main purposes of PIT are (1) recording and pointing to the breadcrumb path, and (2) suppressing incoming redundant interests and data. PIT is essential since breadcrumbs are used in virtually all multi-hop routing protocols. FIB is a name-based routing table designed for a name prefix structure. ICN interests are broadcast to one or more interfaces recorded in the FIB table. The use of broadcast allows ICN to explore multiple caches at no extra cost.

B. Mobile ICN

ICN in mobile environments is a fast emerging research area. Some contributions address VANETs with infrastructure support. For example, J. Wang et al. propose an NDN-based vehicle data collection system in [9] that requires special name prefixes reserved and announced in advance. J. Lee et al. propose a proxy-based scheme to increase the efficiency of mobile retrievals [10].

Ad-hoc ICN without infrastructure support is more challenging. The ICN design follows a late-binding approach in which no name-locator map is established before the interest is sent. This assumes the medium is reliable enough and the "faces" of transmissions are relatively stable, which is not always the case in ad-hoc networks. Data consumer mobility is well served by the ICN design. A data consumer can re-initiate interests to obtain the data from local caches (as opposed from producer) when it relocates. However, handling data producer mobility is difficult. ICN may mitigate effects of data producer mobility by multi-sourcing (from caches), but this issue can become more complicated in vehicular environment. In typical VANET deployment where a node uses only one interface, the interface-selecting design results in all nodes flooding all Interest packets they received. When data producers and relays are moving, resorting to interest flooding may trigger data storms from replicated caches . Therefore, flooding will introduce heavy overhead. One solution is to construct an overlay network on top of IP layer as suggested by S. Oh et al. for a tactical MANET ICN [11].However, implementing point-to-point overlay ICN is costly in VANET on two counts: (1) end-to-end route construction and maintenance between moving overlay nodes induces high control overhead, and (2) overlay design does not exploit wireless broadcast.

For these reasons, most existing ad-hoc ICN research studies propose to build a broadcast-only system to better utilize the wireless channel and minimize control overhead under mobility. In one of the early papers [7], Meisel et al. propose a low-overhead forwarding protocol for MANET ICN. Each node maintains distances to known data object names and node names. The requests are broadcast but propagated to the nearest data holder via shortest path using opportunistic routing [12] based on the destination distances estimated by relays. In [6], Amadeo et al. describe a multi-hop MANET ICN architecture in which the content consumer locates content provider using controlled flooding. Nodes keep track of the content provider IDs and use this information to perform consumer-based provider selection. The transmissions are broadcast but counter-based suppression is applied to reduce unnecessary transmissions. The extended work [3] examines similar system in VANET shows improvements on content delivery over IP-based VANET using AODV in a small scenario in which 5-25 consumers download contents from a road-side unit. However, the counter-based broadcast suppression approach suit low-mobility ad-hoc network but may incur unnecessary transmissions and packet loss due to the interference and inaccurate node distance information under high mobility. In [13], a geo-assisted opportunistic routing and content advertising scheme is proposed. By using geo-coordinates as destinations, this approach eliminates the inaccuracy of distance estimation and provides better eligible forwarder prioritization in opportunistic routing.[14] applies the same geo-assisted opportunistic routing technique and uses last encounter information to discover provider location for unpopular contents. In [5], we proposed a software-defined ad-hoc ICN architecture, which allows the network to adapt to DTN mode when the connectivity is judged disrupted. This paper fills the gap created by DTN mode operations and explains the node-based interest aggregation and hop-by-hop data propagation in more details.

III. System Design

DT-ICAN subsumes both the family of peer-to-peer data dissemination network (e.g. Haggle[15]) in which interests are propagated in an epidemic fashion as well as the family of ICNs in which data is cached as uniquely identifiable chunks [1]. Following the hierarchical naming in [1], files are segmented into chunks. All chunks are named as *applicationID/filename/chunkID*. To leverage the wireless broadcast channel, all communications in this system are broadcast. We summarize the DT-ICAN system architecture in Figure 1. In the following, we explain the major components of DT-ICAN.

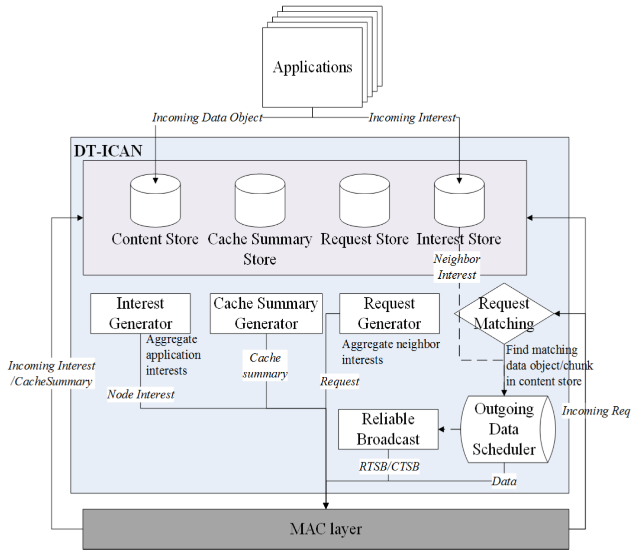

Fig. 1. DT-ICAN System Architecture

A. *Bloomfilter-Based Content Searching*

DT-ICAN reduces the bandwidth consumption by per-node interest aggregation. This is done by splitting the functionality of interest to two periodically broadcast control messages: the *node-interest*, which indicates the data objects a node wants, and the *request*, which represents the data object the node is currently trying to retrieve within one hop. In addition, each node periodically advertises its *cache summary* to assist efficient content requesting. Node-interest, request, and cache summary are all represented by Bloomfilters [16].

1) Node-Interest: Each node aggregates its interests from applications as a node-interest and only periodically broadcast their node-interest to improve the scalability in disruptive networks. The node-interest is compiled in file object basis. In other words, a node indicates the *file object IDs* instead of chunk IDs to speed up the opportunistic retrieval. Instead of instantly flooding node-interests, nodes opportunistically and periodically propagate the node-interests received from others over multiple hops when the bandwidth is sufficient. (III-C)

2) Request: As the encounter intervals in a sparse network can be very limited, the node-interests do not trigger data transmission immediately because otherwise the amount of data transfer triggered can be out of control. We introduce the *request* message, which identifies a subset of data objects that a node is willing to receive at the present time. Note that a request may consist of not only data objects the node itself subscribes but also the data objects others are interested in.

Requests are broadcast only within one hop to retrieve data from neighbours. When a new request comes in, a node examines the request with the names of data in CS, compiles a list of data objects to offer, and then initiates data chunk transmissions in order using the handshake procedure (III-B). The request transmission may be triggered in two conditions:

1) When new contact is discovered: new contact may be discovered when receiving packets from neighbors. If a node detects a new contact, it broadcasts its current request.
2) Periodically: The requests are also periodically broadcast to reflect the changes of the list of data of a node's interest.

Note that the node has the freedom to decide what contents to pull based on the volume of node-interests it receive, the network condition, and its local content prioritization policy. We assume nodes decide the amount of data to retrieve from the newly-encountered neighbor based on the estimated available bandwidth, and aggregates the retrieval in one request to prevent overwhelming redundant data transmissions from multiple caches.

3) Cache Summary: Each node periodically generates its own *cache summary* ane broadcast it to one-hop neighbors. The cache summary includes the data object IDs of fully cached file and chunk IDs of partially cached ones. The cache summaries are leveraged by nodes to prioritize which data chunks to send and request. Without cache summaries, nodes may blindly pull redundant data based on previously received node-interests. All nodes also update neighbors' cache summaries by the data object names carried in the control messages and data overheard.

B. *Request generation and data transfer*

Since node-interest and DTN-request are both represented by Bloomfilters, the most efficient way to generate a DTN-request is by merging a node's own node-interest and the ones received from neighbors. This leads to a node priority-based request generation policy. Namely, the data to include in the request is decided by a node ranking locally computed. The node ranking policy can be flexibly defined. In our implementation, we assume all nodes have the same ranking for simplicity.

When a node receives a request, it finds the requested data objects/chunks available in its content store. This procedure is called *request matching*. A list of available data chunks are compiled, and the node may order the data objects according to node priority and content attributes. Note that this is where the application/content awareness can help improve performance. In our simulation, data objects are prioritized in First-Come, First-Serve order.

When a data transmission is triggered by the request, the available data chunks are sent to the receiving node in random order. The reason of the randomization is to improve the cached chunk diversity of the opportunistic network in case of short encounter durations. An alternative approach is to utilize network coding, which can further improve the performance under high intermittency[17]. A handshake procedure is associated with each data chunk transmission to eliminate redundant transmission in the broadcast network. For each data chunk, the sending node first sends a Request-To-Send-Block (RTSB) carrying the chunk name to the target node. Upon receiving an RTSB, the target node sends a RTSB-Reply, which may accept or reject the block. If the block is rejected, a reject code is carried to indicate one of the three reasons: (1) The chunk is already received, (2) The complete object is already received, and (3) The chunk is being sent by other neighbors. The data is only transmitted if accepted. Once the target node receives the data, it acknowledges by an ACK. Note that all neighbors of the target node also updates their stored cache summaries based on the broadcast RTSB-Reply and ACK.

In case of partial multi-hop connectivity exists, we optimize the multi-hop data forwarding as follows. When a data block is received, a relay propagates the data back towards its original requester(s) by checking pending requests recently received from neighbors. If matches are found, the relay initiates a data transmission for the particular data. In this way, the data is delivered back to the original requestors via the trail of breadcrumbs. To eliminate redundant transmissions, if the data matches multiple interests, only one broadcast data transmission is initiated. This approach achieves the same benefit of per-content interest aggregation as in [1].

C. Node-Interest propagation

While DTN requests are only transmitted within one hop, the node-interests must be propagated so that the relays may start requesting data. Note that node-interests are not instantly flood. Instead, they are broadcast periodically. Note that this may affect system scalability when the number of reachable nodes is large. One potential solution is the following: The nodes periodically broadcast a subset of received node-interests using a given amount of capacity. The rest of the capacity is reserved for data requesting and transfer. The order of node-interest broadcasts are decided using the same node ranking algorithm as that in request generation. Note that while this greedy approach may lead to better scalability, we evaluate the performance using the simple periodic node-interest dissemination as a performance benchmark of the system.

D. Reliable Broadcast

DT-ICAN is an all-broadcast system. The consequence of this design is that there is no MAC layer retransmission support. Therefore, we implement a content-aware reliable broadcast layer to guarantee robust transmission. The reliable broadcast is applied to short control messages destined to particular nodes such as RTSB and RTSB-Reply. It utilizes the data object IDs and node IDs carried in the packets to ensure delivery. An RTSB or RTSB-Reply packet is retransmitted up to certain times if a relay node does not detect a progress is made, by checking the data chunk name carried by incoming RTSB-Reply and Data, respectively.

IV. Evaluation

We compare the performance of DT-ICAN and traditional multi-hop ad-hoc ICN by simulation. We refer to the traditional multi-hop, non-opportunistic ICN as ICN in the following. The simulator is built using QualNet 6.1. The ICN implementation basically follows [1] but with necessary modification for robust multi-hop transmissions as suggested in [13][5]. For fair comparison, the ICN interests are flooded using an opportunistic version[12] of MPR flooding [18][13] to guarantee all reachable nodes will receive the instantly propagated interests; relay nodes are allowed to retransmit a packet by up to 4 times; the relay retransmission interval is 200ms; the requester continuously re-initiate its interest if data is not received every 1 second. In addition, each ICN data chunk is requested as soon as its previous chunk for the data object is received on the requester side to shorten delay. DT-ICAN broadcasts periodic control messages, i.e., cache summary, request, node-interest, every 2 seconds. All other settings for ICN and DT-ICAN are identical. We use IEEE 802.11a MAC/PHY. The data rate is fixed to 36Mbps. The transmission range is approximately 200 meters.

We measure three metrics: average file retrieval rate, average file completion delay, and average per-node traffic. The file retrieval rate is defined as the ratio of number of complete file received to total number of file requested. The file completion delay is the time elapsed since a file is requested/subscribed on the requester side until the file is received completely by the requester. To be precise, for DT-ICAN, the file is requested at the time the data object is added to the node interest of the requester; for ICN, the file is requested at the time the first chunk of the file is requested. The average per-node traffic, which represents the bandwidth consumption, is the average number of bytes each node transmits per second.

A. Synthetic trace in Washington DC map

We first simulate a synthetic trace for better understanding of the performance under various node densities. We downloaded a 2000m by 2000m map of the Washington, DC area made available by the US Census Bureau's TIGER database, and simulating mobility on the map using the Intelligent Driver Model with Intersection Management by VanetMobisim [19]. This model is complete with intersections and stop light rules to simulate realistic vehicular traffic. We generate three scenarios by varying the number of nodes between 50 and 100. The vehicle speed ranges from 11mph to 40mph. The stay time ranges from 5 to 30 seconds. In all scenarios, we randomly selected 9 data source nodes, each publishes a 128KB file. One randomly selected data consumer starts to download all files at 100 second. The simulation time is 1000 seconds. All experiments are repeated 10 times with different random seeds and confidence intervals are reported.

Figure 2 shows the file retrieval rate of the three scenarios. DT-ICAN outperforms ICN as end-to-end paths are not available all-time for all data sources. DT-ICAN's file retrieval rate increases as the number of nodes increases, and is able to retrieve all files in the 100 nodes case. On the other hand, ICN is only able to retrieve around 40% of the files regardless of the node density. Note that even if the number of nodes increase, network partitioning still always exist due to the nature of vehicle movement patterns considering traffic lights, and therefore there are always some nodes not instantly reachable to the requester.

We present the completion delay in figure 3. As expected, the completion delay for both schemes decrease as the number of nodes increases. The reason is that more caches are

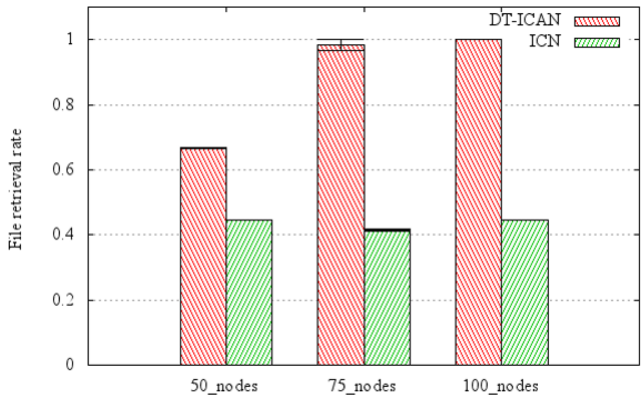

Fig. 2. Washington DC map: file retrieval rate

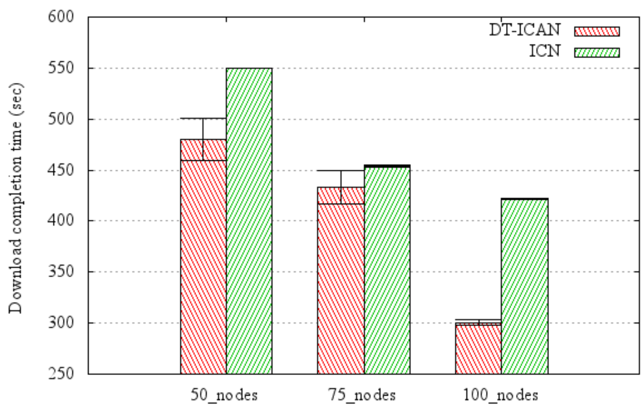

Fig. 3. Washington DC map: file completion delay

available and the chance of node encounters is higher. DT-ICAN achieves up to 30% lower completion delay than that of ICN, as in DT-ICAN multiple caches may request different chunks of the same file simultaneously, leading to better cache diversity than that of ICN. This shows the benefit of data object-level requesting. That is, multiple relays can download different parts of the files for a requester in parallel in DT-ICAN. In ICN, the requests are issued chunk by chunk. Therefore, relays may only request the same chunk(s) at any given time, and hence causes longer delay in a disruptive scenario.

Finally, the average traffic is shown in Figure 4. As expected, the performance gain of DT-ICAN comes with the sacrifice in bandwidth consumption. DT-ICAN has constant traffic as the control traffic, i.e. periodic control messages, persists. However, the aggregated traffic is controlled within 10KBps, which is relatively low in the 36MBps channel.

B. San Francisco taxi trace

In this scenario, we use actual mobility traces of taxi cabs in San Francisco downloaded from Crawdad [8]. The simulation runs in a 5700m by 6600m area with 116 taxies.

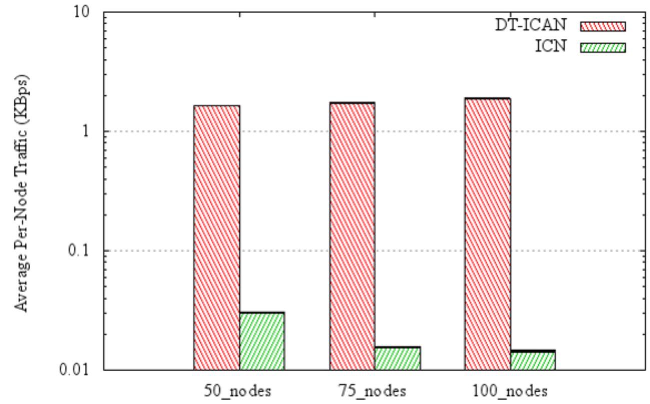

Fig. 4. Washington DC map: average traffic

The simulation time is 3600 seconds. Nine data publishers each publishes a 128KB file. One randomly selected data consumer requests the files. All experiments are repeated ten times. In this scenario, the nodes are sparse because only cab movements are recorded in the trace file. This means that the network's period of disconnection is likely longer than its connectivity. However, end to end path still exists in some cases. This trace very well illustrates the case when ICN is deployed as an overlay.

The simulation results are summarized in Table I. Although the network is very sparse, given enough execution time, DT-ICAN is still able to achieve 100% file delivery, double of ICN's. The average completion time of DT-ICAN is shorter, as expected, and the average traffic is in the same degree as in the previous experiment. Note that although DT-ICAN outperforms multi-hop retrieval of ICN in general, we still find the merit of multi-hop direct requesting. When an end to end path can be found, the completion time can be dramatically shorter if using multi-hop retrieval. This can be seen by observing the minimum download completion time among all files'. When an end-to-end path presents, ICN is able to complete the transfer within a second, while DT-ICAN still needs more time, i.e. in the degree of tens of seconds, for the epidemic interest propagation. This is due to the fact that DT-ICAN must wait for the node-interest to reach the neighbors of data source. Therefore, we suggest an adaptive approach as proposed in [5], in which the retrieval only switches to DTN mode when the end-to-end connectivity is judged inexistent.

We also explored the parameter space of DT-ICAN in this real-world scenario. The most fundamental parameter in DT-ICAN is the interval for periodic control messages. In this experiment, we vary these intervals. The completion delay is presented in Figure 5. As we can see, the lower completion delay is achieved at interval 2 seconds. Shorter intervals cause slightly higher delay due to the queueing time of the messages. Longer intervals also have higher delay since the data transmissions are triggered more slowly. Due to the limited space, the file retrieval rate and traffic are omitted. DT-ICAN is able to retrieve all files with any settings, and the

	DT-ICAN	ICN
File retrieval rate	1 (0)	0.55 (0)
Completion time	821.11 (25.52)	1353.94 (0.61)
Per-node traffic (KBps)	13.43 (2.8)	0.36 (0.02)
Minimum completion time	6.5448 (2.82)	0.2271 (0.03)

TABLE I. TAXI TRACE SIMULATION RESULTS: NUMBERS ARE "AVERAGE (STANDARD DEVIATION)" AMONG TEN RUNS

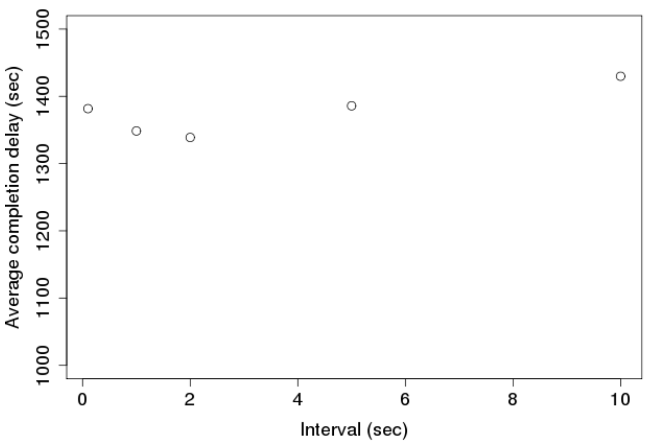

Fig. 5. Varying control message interval: completion delay

average per-node traffic remains close to 6KBps for interval larger than 1 second. Therefore, we judge that the 2 second interval is the optimal setting.

V. CONCLUSION

We introduce DT-ICAN, a disruption-tolerant ICN networking platform aiming at improving efficiency and scalability of ad-hoc ICN in sparse, disruptive scenarios with per-node interest aggregation and epidemic-style propagation. Our simulation results show that in realistic VANET scenario, the epidemic DT-ICAN is necessary to guarantee high file retrieval rate. In addition, the constant overhead caused by DT-ICAN is relatively low, at the degree of several Kbps per node, to the capacity. As multi-hop end-to-end retrieval achieves 100 times lower delay when the two end hosts are connected in real-world trace, we suggest to adopt an adaptive approach in which the system uses multi-hop routing by default, but switches a data object subscription to DTN mode when data holder is unreachable.

ACKNOWLEDGMENT

This work was in part supported by the National Science Foundation under grant no. CNS-1111971001.

REFERENCES

[1] Van Jacobson, Marc Mosko, D Smetters, and JJ Garcia-Luna-Aceves. Content-centric networking. *Whitepaper, Palo Alto Research Center*, pages 2–4, 2007.

[2] Marica Amadeo and Antonella Molinaro. Chanet: A content-centric architecture for ieee 802.11 manets. In *Network of the Future (NOF), 2011 International Conference on the*, pages 122–127. IEEE, 2011.

[3] Marica Amadeo, Claudia Campolo, and Antonella Molinaro. Content-centric networking: is that a solution for upcoming vehicular networks? In *Proceedings of the ninth ACM international workshop on Vehicular inter-networking, systems, and applications*, pages 99–102. ACM, 2012.

[4] Yu-Ting Yu, T. Punihaole, M. Gerla, and M.Y. Sanadidi. Content routing in the vehicle cloud. In *MILITARY COMMUNICATIONS CONFERENCE, 2012 - MILCOM 2012*, pages 1–6, 2012.

[5] Yu-Ting Yu, Chris Tandiono, Xiao Li, You Lu, MY Sanadidi, and Mario Gerla. Ican: Information-centric context-aware ad-hoc network. In *Computing, Networking and Communications (ICNC), 2014 International Conference on*. IEEE, 2014.

[6] Marica Amadeo, Antonella Molinaro, and Giuseppe Ruggeri. E-chanet: Routing, forwarding and transport in information-centric multihop wireless networks. *Computer Communications*, 2013.

[7] Michael Meisel, Vasileios Pappas, and Lixia Zhang. Listen first, broadcast later: Topology-agnostic forwarding under high dynamics. In *Annual Conference of International Technology Alliance in Network and Information Science*, page 8, 2010.

[8] Michal Piorkowski, Natasa Sarafijanovic-Djukic, and Matthias Grossglauser. CRAWDAD data set epfl/mobility (v. 2009-02-24). Downloaded from http://crawdad.org/epfl/mobility/, February 2009.

[9] Jiangzhe Wang, Ryuji Wakikawa, and Lixia Zhang. Dmnd: Collecting data from mobiles using named data. In *Vehicular Networking Conference (VNC), 2010 IEEE*, pages 49–56. IEEE, 2010.

[10] Jihoon Lee Daeyoub Kim Myeong-Wuk and Jang Byoung-Joon Lee. Proxy-based mobility management scheme in mobile content centric networking (ccn) environments. Consumer Electronics (ICCE), 2011 IEEE International Conference on, 2011.

[11] S.Y. Oh, D. Lau, and M. Gerla. Content centric networking in tactical and emergency manets. In *Wireless Days (WD), 2010 IFIP*, pages 1–5, 2010.

[12] Sanjit Biswas and Robert Morris. Opportunistic routing in multi-hop wireless networks. *ACM SIGCOMM Computer Communication Review*, 34(1):69–74, 2004.

[13] Yu-Ting Yu, Xiao Li, Mario Gerla, and MY Sanadidi. Scalable vanet content routing using hierarchical bloom filters. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International*, pages 1629–1634. IEEE, 2013.

[14] Yu-Ting Yu, Yuanjie Li, Xingyu Ma, Wentao Shang, MY Sanadidi, and Mario Gerla. Scalable opportunistic vanet content routing with encounter information. In *2nd International Workshop on Vehicular Communications and Applications (VCA13)*, 2013.

[15] James Scott, Jon Crowcroft, Pan Hui, Christophe Diot, et al. Haggle: A networking architecture designed around mobile users. In *WONS 2006: Third Annual Conference on Wireless On-demand Network Systems and Services*, pages 78–86, 2006.

[16] Burton H Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.

[17] Joshua Joy, Yu-Ting Yu, Mario Gerla, Mark-Oliver Stehr, Sam Wood, and James Mathewson. Network coding for content-based intermittently connected emergency networks. Mobicom Demo '13, 2013.

[18] Amir Qayyum, Laurent Viennot, and Anis Laouiti. Multipoint relaying for flooding broadcast messages in mobile wireless networks. In *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on*, pages 3866–3875. IEEE, 2002.

[19] Jérôme Härri, Fethi Filali, Christian Bonnet, and Marco Fiore. Vanetmobisim: generating realistic mobility patterns for vanets. In *Proceedings of the 3rd international workshop on Vehicular ad hoc networks*, pages 96–97. ACM, 2006.