Enhanced Data Delivery Framework for Dynamic Information-Centric Networks (ICNs)

Fadi M. Al-Turjman[1] and Hossam S. Hassanein[2]

[1]School of Engineering
University of Guelph
Guelph, Ontario, Canada N1G 2W1
fadi@uoguelph.ca
[2]School of Computing
Queen's University
Kingston, Ontario, Canada K7L 3N6
hossam@cs.queensu.ca

Abstract— **In this paper, we present an Enhanced 2-Phase Data Delivery (E2-PDD) framework for Information-Centric Networks (ICNs), focusing on efficient content access and distribution as opposed to mere communication between data consumers and publishers. We employ an approach of growing eminence, where requests are initiated by consumers seeking particular services that are data-dependent. High-level Controllers (HCs) receive the consumers' requests and issue queries to a multitude of data publishers. The publishers in our topology include a wide variety of ubiquitous nodes that could be either stationary or mobile, operating under different protocols. In order to consider fundamental challenges in ICNs such as node mobility and data disruption, our E2-PDD framework employs Low-level Controllers (LCs) that act as moderators between the HCs and the data publishers, executing data queries for a top tier and replying back with a set of candidate rendezvous points obtained from a bottom tier. The HCs maximize selection *based on the nearest rendezvous*. Extensive simulation results have been used to evaluate our E2-PDD framework in terms of key performance metrics in ICNs viz., average in-network delay, and publisher load, given different mobility pause time durations and data consumers' densities.**

Keywords- Information-centric networks; disruption-tolerance; 1-phase data delivery;2-phase data delivery.

I. INTRODUCTION

The majority of Internet usage today consists of data retrieval and service admission, where the user is interested in the content regardless of its location. Examples include obtaining news headlines or access to private bank accounts irrespective of the machine where the required data or service resides. In fact, the current Internet paradigm, which has been built around ensuring end-host connectivity, is being challenged as the Internet is becoming more of a medium for publishing and retrieving information regardless of the publisher or the networking protocol it runs [1]. This view is behind a variety of information-centric solutions where the terms *information*, *content*, and *data* are used interchangeably.

The demand for scalable and efficient distribution of data has motivated the development of an architectural approach known as Information-Centric Networking (ICN) that is based on Named Data Objects (NDOs) such as web pages, multimedia documents, personal metrics, or other forms of content [2]. In general, ICNs aim at efficient access and distribution of content in addition to being adaptable to disconnections and service disruptions. In-network storage (caching) is leveraged to provide better transfer service. Communication is based on a publisher-consumer relation where publishers make NDOs available to requesting consumers by publishing the objects. Hence, data and queries are self-addressable and self-routable.

The aforementioned description includes features from two broader networking concepts; the first is the Internet of Things (IoT) which defines interconnected settings where all nodes (things) are uniquely addressable and traceable [3]. The second is Delay/Disruption-Tolerant Networking (DTN) in which no end-to-end link exists; transferred data is to be cached until a suitable forwarding opportunity arises in a Store-Carry-Forward (SCF) fashion [4]. We argue that if privately-owned wireless devices such as smart phones, tablet PCs and medical sensory gadgets are to be considered as data publishers in an ICN network, then their transmission and routing resources would be heavily utilized in a manner that will eventually prevent their owners from participating in the ICN. Moreover, considering the mobility of these wireless devices and the data disruption it causes, due to continually changing locations, would result in depletion of the NDOs. Therefore, it is necessary to relieve such data sources (publishers) from this load which is expected to be enforced by the data requesters (consumers) in a typical ICN network.

To this end, we summarize our contributions as follows:

- We introduce E2-PDD, an Enhanced 2-Phase Data Delivery framework for ICNs that response to data retrieval requests by examining both data publishers and/or intermediate nodes in the vicinity while adhering to mobility, and delay constraints.
- Our E2-PDD framework incorporates a multi-tier architecture that introduces new entities between the data publishers and consumers called Low-level Controllers (LCs). These LCs are to organize the data flow between the top and bottom tiers of the architecture and relieve the High-level Controllers (HCs) from undesired overhead communications.
- Our E2-PDD framework relies on ICN-specific design factors such as node mobility, data caching, and delay

constraints in providing the most appropriate Rendezvous Points (RPs) from which data consumers can pick up their requested data.

- An ICN-specific disruption metric is used to predict data delivery time. This metric is significant in providing the most appropriate RPs in case of delay-sensitive data.

The remainder of this paper is organized as follows: Section II surveys related work. Section III describes our system models. Section IV introduces our and problem statement and the proposed E2-PDD framework. Section V evaluates the performance results of our approach in comparison to prominent data delivery approaches in ICN. Lastly, Section VI concludes this paper.

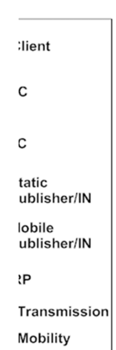
II. Background

Data delivery in ICNs can be categorized into either: 1-Phase Data Delivery (1-PDD) [6][8], or 2-Phase Data Delivery (2-PDD) [5][7]. In 2-PDD, a dedicated content mediation plane (CMP) is used to address the requested NDO given the current network conditions, while in the 1-PDD, there is no intermediate CMP for such a hierarchical control. Accordingly, it's called 2-PDD, where the data delivery process passes in to two main phases. In the first phase, the NDO is mapped to the most suitable locator (e.g., an IP address using a DNS server in CMP). In the second phase, the locator is used to locate the NDO and deliver it to the requester (consumer). For example, in the Data-Oriented Network Architecture (DONA) proposed in [5], NDOs are published into the network by the publishers. Nodes that are authorized to serve data, register to the resolution infrastructure consisting of resolution handlers (RHs) which forms the CMP. Requests are routed by name toward the appropriate RH. Data is sent back in response, either through the reverse RH path enabling caching, or over a more direct route implying a higher transmission load on the source node. Similarly, in Publish-Subscribe Internet Routing Paradigm (PSIRP) [7], NDOs are also published into the network by the NDO sources. Receivers can subscribe to NDOs. The publications and subscriptions are matched by a rendezvous system at the CMP. The subscription request specifies the scope identifier (SI) and the rendezvous identifier (RI) that together name the desired NDO. The identifiers are input to a matching procedure resulting in a forwarding identifier (FI), which is sent to the NDO source so that it can start forwarding data.

Unlike the 2-PDD, in 1-PDD an NDO is located and delivered based on Data Advertisement (DA) packets. Each node receiving a DA packet stores information about such advertised data (or NDO) to locate it by this node itself or by another node (consumer). Thus, when a data request is received at a node, one of the following scenarios applies: 1) If this node has the NDO, it delivers the requested NDO back to the consumer, 2) If it knows location of the NDO based on a previous DA packet, it informs the consumer about this NDO location, or 3) If it has no information about the requested NDO, it forwards the request to the following unvisited node in the network. For example, in Content-Centric Networking (CCN) [6], NDOs are published at nodes, and routing protocols are employed to distribute information about NDO location. Routing in CCN can leverage aggregation through a hierarchical naming scheme. NDO is achieved through public key chain based on the naming hierarchy, or information provided by a friend. Requests (interest packets) for an NDO are forwarded toward a publisher location. CCN supports on-path caching: NDOs a CCN router receives (in responses to requests) can be cached so that subsequent received requests for the same object can be answered from that cache. Also, there are attempts towards proposing ICNs that support both 1-PDD and 2-PDD. For example, the Network of Information (NetInf) project [8] offers two models for retrieving NDOs, via name resolution (i.e., 2-PDD) and via name-based routing (i.e., 1-PDD), thereby allowing adaptation to different network environments. In NetInf, depending on the model used in the local network, data sources publish NDOs by registering a name/locator binding with a name resolution service (NRS), or announcing routing information in a routing protocol. A NetInf node holding a copy of an NDO (including in-network caches and user terminals) can optionally register its copy with an NRS, thereby adding a new name/locator binding. If an NRS is available, a receiver can first resolve an NDO name into a set of available locators and can subsequently retrieve a copy of the data from the "best" available source(s).

The 2-PDD approach supports scalability with adopting hierarchical structures and binding aggregation. However, it suffers from problems related to the Name Resolvers (NRs). Such NRs can be attacked and used to present false information to the network and consumers. NRs are susceptible to denial of service attacks as well. On the other hand, the 1-PDD approach is more flexible and can adapt to the varying network conditions. However, it suffers from scalability problems because of having to disseminate information about owned objects widely in the network which, in turn, results in BW wasting problems and inflations in the routing and interest tables.

Meanwhile, both 1-PDD and 2-PDD approaches aim at considering node mobility and disruption tolerance, which are



rk.

fundamental issues in ICNs. Mainly, the selection of the most appropriate data delivery approach depends on the level of knowledge available regarding the networks' topology. In an ICN setting, such knowledge is either partial or absolutely absent. Typically, if the nodes possess no routing information about their neighbors, then they implement broadcasting protocols that blindly forward copies of their packets to any neighbor they encounter. This ranges from full network flooding [11], [12] to partial flooding [13]. Network coding [14] was further proposed to improve the performance of ICN flooding. Blind routing may achieve a high delivery ratio provided enough storage and energy resources. Yet, it burdens the node buffer and inefficiently utilizes the contact duration. Thus, broadcasting protocols are not favored for their costly operations. Topology control [15] is proposed to achieve efficient broadcasting with low interference and low energy consumption. Alternatively, if some knowledge of the nodes' mobility patterns or routing history is available, then guided routing [16]-[18] is applied. Protocols in this class exchange routing tables to assign weights to nodes/links based on information collected from the network. This information may be related to the contact between nodes or their location or mobility. For instance, history-based probabilistic routing in [16] and [17] estimates a delivery predictability metric that is strengthened each time particular nodes meet. In addition, social-based routing [18] is particularly useful in cases where human mobility traces are involved in the routing process.

As will be mentioned in Section IV, the algorithm we present here is based not only on avoiding the exhaustive exchange of routing tables between neighboring nodes, but also on providing available information about the mobile data sources (publishers) through a new entity called Low-level Controllers (LC). It should be mentioned, also, that data delivery (routing) in ICN has not been addressed in existing Delay/disruption-Tolerant Network (DTN) schemes. Other DTN delivery schemes are based on utilizing the mobility patterns of nodes in the topology. We particularly note the Data MULEs (DMs) approach introduced in [19]. DMs were defined as mobile nodes with arbitrary mobility patterns and equipped with large storage capacities and renewable energy sources. This concept is further applicable nowadays due to the wide spread of wireless smart devices. The proposal in [19] consisted of a three-tier architecture (sensor nodes, DMs and access points) and is supposed to connect sensors at the cost of high latency. Unlike this proposal, our E2-PDD approach aims at utilizing LCs in order to provide more appropriate rendezvous points from which the consumer can collect data rather than waiting for a mobile DM to carry data.

In general, our approach aims at predicting candidate rendezvous to meet requested data while considering predetermined node mobility. This results in more disruption-tolerance and on time data delivery.

III. System Models

In this section we describe the network, disruption, cache and data delivery models that are the basis of our E2-PDD scheme. In addition, we present the assumptions related to it.

A. Network Model

The components comprising our E2-PDD multi-tier architecture are listed as follows:

- Data *Publishers* that represent node capable of publishing an NDOs. According to the E2-PDD scheme, *publishers* may be either static or mobile. They include sensors, RFID tags, Tablet PCs, smart phones and any ambient node that may provide NDO content in any form.
- Intermediate Nodes (IN) are in-network devices that perform processing/caching for the NDOs such as routers, switches, relays, etc. However, they can't publish data by themselves.
- Consumers (Clients) are the data requesters in an ICN setup. This component can be in reality a hand held smart phone, laptop, smart car, etc.
- Rendezvous Points (RP) are INs in the ICN network domain that acts as a shared point for a multicast shared request. Any number of INs can be configured to work as RPs and they can be configured to cover different group ranges of consumers. Every HC within our E2-PDD framework must be able to map a particular multicast group address to the same RP.
- High-level Controllers (HCs) that initiate data requests based on the queries received by consumers. Based on the attributes of the data request, an HC will set a delay sensitivity level to the requested data represented by D_{th}. In addition, it will decide the specific geographical region to which the request will be forwarded, based on the consumer location and requested data sources.
- Low-level Controllers (LCs) that serve as moderators between the HC and the Publisher/IN nodes at the lower tier of the architecture. LCs are assumed to cover all Publisher/IN in the network. LCs are responsible for replying to HC requests by the required information about the data holders (either a publisher or an IN) and delivering it to the HC to make decision on which RP to meet.

According to the description above, LCs are to relieve *publishers* from any transmission load. Moreover, the LCs nominate candidate RPs from the perspective of the underlying caching pool in the ICN since each LC is immediately connected to the set of publishers/INs within its interrogation zone. As for the HCs, they provide consumers with the most appropriate RP choice from their perspectives based on a set of recommended RPs by the LCs and the consumer location within the network. Fig. 1 shows a scenario where all the aforementioned components cooperate to provide an HC with a list of candidate RP for the requested data by the consumer. If the requested data is found at any of the publishers/INs, then the corresponding LC replies immediately to the HC with the most appropriate RP candidate. This process will be further elaborated upon in Section IV.

B. Disruption Model

Disruptions, disconnections and delays in ICNs may occur when a popular NDO source links to a smaller site, causing a massive increase in traffic in an effect known as *slashdotting*. End-to-end communication to source nodes is often difficult to achieve in challenged networks, with sparse connectivity, node mobility, and disruptions. Due to dense network topologies associated with ICNs, a relatively long multi-hop path consisting of heterogeneous devices can easily exist between the Publisher/IN and the corresponding RP. In addition, these devices might not have an immediate communication link with the next hop which causes extended delay periods while holding data till the communication link is available; either because it is a mobile node with a limited speed s, or because it's a static node buffering data until a mobile collector will arrive to pick it up. Thus, the disruption/delay components we have to consider are dominated by the queuing/propagation delay ψ, and the holding delay ϵ. These components are extremely dependent on the relaying hops count and the node mobility patterns. Accordingly,

$$\psi \propto N \tag{1}$$

where N is the total number of hops between the packet's source and the consumer. And the holding delay ϵ is determined based on the following:

$$\epsilon \propto (\frac{d}{s} + P) \tag{2}$$

where d is the distance separating a mobile data holder or collector from the next hop on path towards the data consumer, and P is the pause time this mobile node would experience during its physical motion. Hence, we define a discretized disruption step D_{LC_i}, which is the delay a packet would experience until it reaches the data consumer via a Low-level Controller i, as follows:

$$D_{LC_i} = \lceil \psi + \epsilon \rceil \tag{3}$$

Thus, we define a normalized D'_{LC_i}as

$$D'_{LC_i} = \frac{D_{LC_i}}{D_{max}} \tag{4}$$

where D_{max} is the maximum expected delay. Based on the aforementioned ICN delay and mobility constraints, choosing the right data gathering strategy has a great potential, and this is the main motivation for this work.

C. Data Delivery Model

We build on the previous data collecting scenarios as well to define our frameworks data delivery models. Delivery is based on required data's delay constraints. In order for publishers/INs to deliver their delay-tolerant data to the data consumers, they need to have a path to at least one RP. To do so, data publishers/INs broadcast their identity at the deployment stage and each node keeps a record of the next hop towards some RP. Each publisher/IN n_i has a Rendezvous Point Record (RPR_i) which has the following fields:

- *id*: the id of the RP to which delay-tolerant data will be sent.
- *Next_hop*: a neighbor of n_i which is used as a next hop towards the RP.
- *Number_of_hops*: the number of hops to the RP.

These records are used to determine the route from a publisher/IN to a RP as will be shown in Section IV.

D. Caching Model

Our E2-PDD scheme adopts a caching model similar to what was proposed in [9]. When a LC entity receives a request for content from an HC, it does one of two actions: (i) if it has the data immediately available either from the static publisher under its coverage or cached from a previous encounter with a mobile publisher, then it can respond with the content directly, or (ii) if it does not have the content cached, the LC requests the content from mobile publishers and then caches the content when this request is fulfilled. This caching is universal in three ways. First, it applies to content carried by any protocol, not just content carried by a specific protocol (e.g., HTTP). Proposed E2-PDD framework thus provides a single uniform caching paradigm that underlies all content delivery. Second, it applies to all content from all users, not just content from content providers who have contracted for the service. This democratizes content delivery. Third, it is implemented by all ICN nodes rather than just a few specialized caches, making caching in E2-PDD pervasive.

We note that requesting data from mobile publishers involves estimating the delay caused by the publisher's mobility until it inters the range of the LC (holding delay). This would be based on an aggregate history function that registers the mobility pattern of each encountered publisher and the type of data it usually provides. If the LC is to wait for a mobile publisher, it should notify the HC which will compare this to the delay limit put on the requested data and either accept or reject the LC's offer.

IV. Problem Statement and E2-PDD Framework

According to the abovementioned ICN system models, the proposed framework addresses two significant problems:

1) Finding a mechanism that manages the relationship between the LCs and the HCs at the top-tier of the delivery model. This mechanism is supposed to help each HC to pick the right rendezvous point according to data sources' information available at the corresponding LC for each data request.
2) Finding the data delivery infrastructure for the lower-tier of the model that supports nodal-mobility and disruption-tolerance in an ICN.

Henceforth, the targeted E2-PDD framework aims towards the following:

Given the two CMPs in an ICN; HCs are to find the best LCs to cater for their data requests within specific geographical vicinities while considering node mobility and network disruptions.

The following two subsections explain our framework to tackle each of the aforementioned problems.

A. High-level Data Delivery

We specify two algorithms at HCs and LCs. These algorithms detail how an HC will send a data request and how it will process the parameters received from acknowledging LCs into candidate rendezvous points to retrieve data. The algorithm at the LCs sets the parameters to be included in a reply to an HC's request and the how each LC will determine its requested data location according to the availability of its publishers and caching devices. Algorithm 1 specifies the steps of a query issued by an HC seeking data (for a consumer) from any LC in the set $\{LC_0, \dots, LC_n\}$. This set is determined according to the geographical vicinity (GV) of the request (line 9). As mentioned earlier, the HC waits for the LC acknowledgements and bases its selection decision on the returned parameters (line 11). Then, the score of each LC is calculated according to the most appropriate RP (lines 14-16). The HC finally choose the appropriate LC (line 17). Algorithm 2 shows how a LC responds to an HC data request. The LC validates the request on two levels. Firs it checks if the requested data is available within INs immediately under immediate coverage (lines 7-10). If this was the case, then the delay component (D'_{LC_i}) in Eq. (3) is expected to be zero and the whole delay factor is hence drastically lower than the following scenarios (lines 11-15) in which the LC checks with mobile publishers/INs within the specified GV. In either case, whenever the data is found, LC-corresponding ϵ and D'_{LC_i} are

Algorithm 1: HC seeking LC to provide an RP.

Function ***HC (Consumer_Req.x)***
Input
Consumer_Req.x: A consumer x data request to be located by this HC.
Output:
LC_i: A selected $LC_i \in \{LC_0, \dots, LC_n\}$ with the nearest RP to receive data from.
Begin
1. **Set** D_{th} // *Data latency constraint*
2. **Initialize** *GV* // *GV is the* geographic vicinity
3. **Set** *{LC}=Φ* //where *{LC}* set of replying LCs
4. **Request** *Data* **from** *LCs in GV*
5. **While** an *Ack* is received from LC_i
6. **If** $D'_{LC} \leq D_{th}$ **then**
7. **Add** LC_i to $\{LC\}$
8. **End**
9. **End**
10. **For each** $LC_i \in \{LC_0, \dots, LC_n\}$ **do**
11. **Check** if it has *the nearest RP.*
12. **End**
13. **Select** LC_i with *the nearest RP.*
14. **End**

assigned to it according to Eqs. (2) and (3) and are included in an acknowledgement to the HC request (lines 11 and 13). Once the HC accepts the offer, the data is sold (lines 17-19).

B. Low-level Data Delivery

In order for Publishers/INs to deliver their delay-tolerant data to the consumer, they need to have a path to at least one RP (Fig. 1). Data is stored in the RP until it is picked up by the consumer. This has to be managed by the LCs in our E2-PDD framework. Delay is dominated here by the physical motion of the publisher/IN (see Algorithm 2) and has two factors. The first is the speed of the publisher/IN. The second is the in-network delay until data is received by the consumer as described in the abovementioned system models. While controlling the speed of the mobile publishers/INs is out of the scope of our work, we can minimize the travelled distance to get data by sending the data to an RP that is very close to the consumer. Each IN n_i maintains an RP Record (RPR_i) with the structure previously described in Section III-C. In

Algorithm 2: LC reply to HC query request

Function ***LC (Request)***
Input:
Request: A data request from HC_i.
Output:
RA: Request answer that could be an Ack to the Data request including nearest rendezvous point, and expected delay.
Begin
1. **Initialize** *list of static and mobile* ***publishers/INs*** *in GV*
2. **If** requested *Data* is available at static publishers/IN **then**
3. **Set** $\epsilon = 0$
4. **Set** D'_{LC_i} //based on Eq. (3).
5. **Return** (*RA=Ack*) with D'_{LC_i}
6. **Else if** requested data is at mobile publishers/IN **then**
7. **Set** ϵ & D'_{LC_i} //according to Eq. (2) & (3).
8. **Return** (*RA=Ack*) with d_{LC_i}, and D'_{LC_i}
9. **Else**
10. **Ignore**
11. **If** this *LC* is selected by HC_i **then**
12. **Return** (*RA= nearest RP*)
13. **End**

consequence, for delivering delay-tolerant data, Algorithm 3 describes the process of setting the RPRs of all INs, assuming that each IN uses the nearest RP. Note that this process will construct a tree for each RP; the tree of a RP n_i is rooted at n_i and involves all INs whose nearest RP is n_i. RPRs will be identified at the initialization of the network.

V. PERFORMANCE EVALUATION

The proposed E2-PDD approach is simulated in this section using MATLAB against the other two approaches which are dominating all data delivery approaches in the literature; the 1-PDD and the 2-PDD. These approaches represent our comparison baseline in this study. In fact, simplified variations of these approaches were widely studied in the literature [11]-[19]. We have constructed a packet-level simulator that allows us to observe and measure these approach's performance under a variety of circumstances. It simulates multi-hop wireless networks while considering practical assumptions in physical, data link and Medium Access Control (MAC) layer models. For the MAC layer protocol it assumes a Distributed Coordination Function (DCF) of IEEE 802.11 [20] for wireless LANs. It uses Request-To-Send (RTS) and Clear-To-Send (CTS) control packets for "unicast" data transmission to a neighboring node. Data packet transmission is followed by an ACK. "Broadcast" data packets and the RTS control packets are sent using physical carrier sensing. An unslotted CSMA technique with collision avoidance (CSMA/CA) is used to transmit these packets. The radio model is implemented based on the probabilistic communication model proposed in [9] in order to determine connectivity between the network wireless nodes. The shortest path routing protocol is assumed in both 1-PDD and 2-PDD in order to guarantee their efficiency in terms of delay. It "detects" all data packets transmitted or forwarded, and "responds" by invoking routing activities as appropriate. The request packets are treated as broadcast packets in the MAC. Response and data packets are all unicast packets with a specified neighbor as the MAC destination. It detects link breaks using feedback from the MAC layer. A signal is sent to the routing layer when the MAC layer fails to deliver a unicast packet to the next hop. Simulated network nodes maintain a send buffer (cache) of 64 packets, which contain all data packets waiting for a next hop neighbor. All packets (both data and routing) sent by the routing layer are queued at the interface queue until the MAC layer can transmit them. The interface queue is maintained as a priority queue with two priorities each served in FIFO order. Routing packets get higher priority than data packets.

Our simulations are run using ad hoc networks of varying number of nodes under a nominal bit rate of 2 Mbps. Mobile publishers/INs move with a random speed that is uniformly distributed between 0 and 20 m/sec. Each mobile publisher/IN starts its journey from a random location to a random destination. Our simulator varies the mobility pattern by varying the pause period. The smaller the pause period, the higher the mobility, and, vice versa for the lower the mobility. This implies that varying the length of the pause period is equivalent to varying the mobility model. We assume the random waypoint mobility model [21] in a grid rectangular field. Two field configurations are used: (i) 1500m x 600m field with 25 consumers out of 500 total nodes and (ii) 2500m x 600m field with 250 consumers out of 500 total nodes. Simulations run for 140 minutes. For each network size, we test 20 instances and take the average.

A. Performance Metrics

To compare the performance of the proposed E2-PDD approach, the following three performance metrics are used:

1) Average Request per Publisher (ARP), measured in request per hour (req/hr) and it represents the average load per publisher in an ICN network. Due to not only in-network caching but also to data delivery approach which participate in disseminating/distributing the in-network data, the publisher load can be reduced. Thus, bandwidth and computing resources of the publisher can be saved.
2) Average Delay, measured in seconds (*sec*) and is defined as the average amount of time required to deliver a data unit to the consumer.
3) Packet Delivery Ratio (PDR), which is the average percentage of transmitted data packets that succeed in reaching the consumer reflecting the effect of delay on data delivery over the utilized data delivery approach.

While studying these performance metrics, we vary three main parameters:

Algorithm 3: Disruption-tolerant routing record

```
Function IN( )
For each IN n_i do
    if n_i is a RP then
        RPRi.id=i;
        RPRi.next hop=i;
        RPRi.number of hops=0;
        broadcast RPi to all neighbors of n_i ;
    else
        RPRi.number of hops = N + 1;
    end
end

when a IN n_i receives a broadcasted RPRj:
    if RPRj.number of hops + 1 < RPRi.number of hops then
        RPRi.number of hops = RPRj.number of hops + 1;
        RPRi.id = RPRj.id;
        RPRi.next hop = j;
        broadcast RPRi to all neighbors of ni ;
    end
end
```

1) The size of the network in terms of total consumer count and the scalability of the exploited data delivery scheme. In addition, it reflects the ICN application's complexity.
2) Pause time for the mobile data publishers and INs as a major delay factor, and its measured in seconds (*sec*).
3) Simulation time, measured in minutes (*min*) and it shows how long it takes to reach steady state in terms of publisher load and delay.

B. Simulation Results

We present simulation results covering performance variations with respect to two primary factors stemming from our E2-PDD framework. These factors are: Publisher load in an ICN (represented here in ARP), and delay (seconds). Figs. 2-7 show our results. In the following subsections, we detail our observations regarding these results.

1) Performance with respect to Publisher Load

From the perspective of publisher load, there are significant gains from E2-PDD as it satisfies the majority of requests at network-level which reduces the publisher load by at least 30% (Fig. 2 and 3). Also in Fig. 2, we notice a remarkable decrement in terms of publisher load while 2-phase data delivery approaches are applied and pause time is incremented. This can be returned to the centralized control of the network in such data delivery category where mobility patterns speak. However, even with a very short pause time periods, our E2-PDD approach proves its efficiency due to the considered RPs which position requested data according to data consumers rather than data sources as in regular 2-PDD approaches. Furthermore, our E2-PDD framework has a significant advantage over the other two categories of data delivery in ICNs. Fig. 3 shows that as nodes' caches become gradually filled after the simulation starts, the traffic to the server drops as more NDOs can be fetched from the in-network cache. We can see that E2-PDD saved more traffic than 1-PDD, while 2-PDD has the worst performance, with traffic reduction of 30%, and 63%, respectively. There is a drastic decrease in server traffic at the beginning of the curves of E2-PDD and 1-PDD as they rely on using the most popular routes for the consumers in delivering data.

2) Performance with respect to PDR

Fig. 4 shows the effects of the different data delivery approaches on an ICN network performance in terms of the delivery ratio (PDR). The packet delivery ratios of the three approaches increase as the pause time increases. The reason is that the stronger route stability is the higher packet delivery ratio. However, E2-PDD is the best, and 1-PDD and 2-PDD are the worst because they don't consider the RP concept in a centralized way as opposed in the E2-PDD approach. Where E2-PDD approach can detect the path that is more likely to be disconnected soon based on the available information at the LCs. This makes the percentage of lost packets very negligible while applying the E2-PDD.

3) Performance with respect to Delay

Fig. 5, 6 and 7 presents the impact of data delivery approach on the network delay. Fig. 5 shows the effect of mobility pattern on the data delivery delay while considering a small size network (of 25 consumers). Although our E2-PDD approach outperforms the other approaches, the three of them are so close in delay results. Nevertheless, a significant difference appears in Fig. 6 while considering large size networks, where half of its nodes are data consumers. This is a typical case in any future network, such as ICNs [2]. This figure shows a great effect of the pause time on data delivery delay while applying regular 2-PDD approaches. However, this effect is reduced significantly while applying our E2-PDD approach. As the proposed E2-PDD approach is not a pure centralized approach, where LCs work on providing a distributed data disseminations in the low-tier of the network. This makes the HCs of the proposed framework more aware of the publishers/INs mobility patterns in an ICN network. Finally, Fig. 7 shows the delay performance while applying our E2-PDD on short and/or long run. Obviously, the E2-PDD approach strength in terms of delay is major on the long run. That is because the data has not yet been cached on in-network nodes and it takes a while to differentiate from the other approaches due to the overhead in communications at the early stages of the network operation.

VI. Conclusion

In this paper we introduce E2-PDD – an Enhanced 2-Phase Data Delivery framework for ICN networks. Our framework is based on a multi-tier architecture that caters for heterogeneous (mobile/static) data publishers and/or intermediate nodes. According to our framework, High-level Controllers (HCs) at the top of the architecture receive consumer queries and initiate data requests. E2-PDD implements algorithms that realize delay-tolerant data requirements. It provides a dynamic two-tier data delivery that acts based on both ends of the publisher-consumer chain. At the top tier, E2-PDD maximizes the consumer's gain according to the delay limit by providing the nearest Rendezvous Point (RP) to meet the requested data. At the bottom-tier, it maximizes the gain of the publishers by reducing their load. We provide simulation results showing the efficiency of our framework when compared to two dominant delivery approaches. Our simulation results show that the E2-

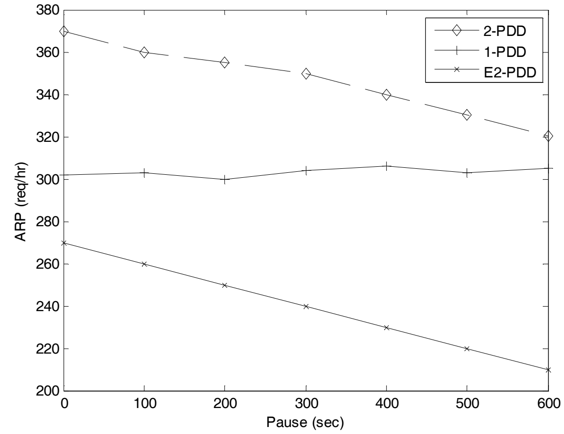


Fig. 2. Publisher load vs. pause time (with 250 consumer).

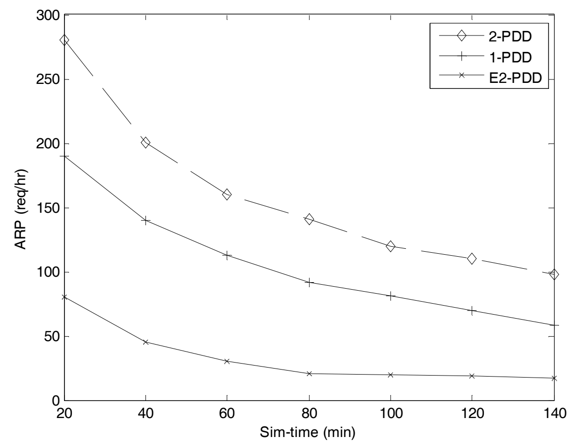


Fig. 3. Publisher load vs. simulation time (with 250 consumer).

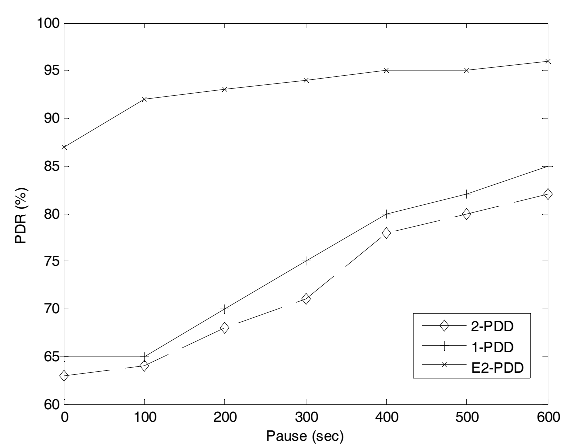


Fig. 4. Avg. packet delivery ratio vs. pause time (with 250 consumer).

PDD framework exhibits superior performance in terms of publisher load, end-to-end delays and packet delivery ratios for different network sizes, and mobility patterns.

I. ACKNOWLEDGEMENT

This research is funded by a grant from the Ontario Ministry of Economic Development and Innovation under the Ontario Research Fund-Research Excellence (ORF-RE) program.

REFERENCES

[1] L. Atzori, A. Iera, G. Morabito, The Internet of Things: A survey, *Computer Networks*, vol. 54, no. 15, pp. 2787-2805, October 2010.

[2] B. Ahlgren, et al., “A survey of information-centric networking” in *IEEE Communications Magazine* , vol. 50, no. 7, pp. 26-36, July 2012.

[3] INFSO D.4 Networked Enterprise & RFID INFSO G.2 Micro & Nanosys., in co-operation with the Working Group RFID of the ETP EPOSS, Internet of Things in 2020, Roadmap for the Future. Ver. 1.1, May 2008.

[4] F. Nordemann, R. Tonjes, “Transparent and autonomous store-carry-forward communication in Delay Tolerant Networks (DTNs),” in *Int. Conf. on Computing, Networking and Communications (ICNC),* pp.761-765, Jan. 30-Feb. 2, 2012.

[5] T. Koponen et al., “A Data-Oriented (and Beyond) Network Architecture,” *Proc. SIGCOMM ’07*, Kyoto, Japan, pp. 27–31, 2007.

[6] V. Jacobson, M. Mosko, D. Smetters, and J. Garcia-Luna-Aceves. Content-centric networking. Whitepaper, Palo Alto Research Center, Jan. 2007.

[7] M. Ain et al., “D2.3 – Architecture Definition, Component Descriptions, and Requirements,” Deliverable, PSIRP 7th FP EU-funded project, Feb. 2009.

[8] B. Ahlgren et al., “Second NetInf Architecture Description,” 4WARD EU FP7 Project, Deliverable D-6.2 v2.0, Apr. 2010, FP7-ICT-2007-1-216041- 4WARD / D-6.2. http://www.4ward-project.eu/

[9] F. Al-Turjman, A. Al-Fagih, W. Alsalih and H. Hassanein, “A delay-tolerant framework for integrated RSNs in IoT,” *Computer Communications*: 10.1016/j.comcom.2012.07.001. Available online 11 July 2012.

[10] A. Ghodsi, et al., “Information-Centric Networking: Seeing the Forest for the Trees,” *in ACM HotNets-X*, Nov. 2011.

[11] K. A. Harras, K. C. Almeroth and E. M. Belding-Royer. Delay Tolerant Mobile Networks (DTMNS): Controlled Flooding in Sparse Mobile Networks, in: Lecture Notes in Computer Science, vol. 3462, 2005, pp. 1180-1192.

[12] T. Spyropoulos, K. Psounis, et al. Spray and Wait: An Efficient Routing Scheme for Intermittently Connected Mobile Networks. *in Proc. of WDTN*, 2005.

[13] A. Vahdat and D. Becker. Epidemic Routing for Partially Connected Ad Hoc Networks, Technical Report CS-200006, Department of Computer Science, Duke University, Durham, NC, 2000.

[14] Y. Wang, S. Jain, M. Martonosi, and K. Fall. Erasure-Coding Based Routing for Opportunistic Networks. *in Proc. of SIGCOMM*, 2005.

[15] K. Miyao, H. Nakayama, N.Ansari and Nei Kato. LTRT: An Efficient and Reliable Topology Control Algorithm for Ad-Hoc Networks. *in IEEE Transactions on Wireless Communications,* vol. 8, no. 12, pp. 6050-6058, 2009.

[16] A. Lindgren, A. Doria and O. Schelen. Probabilistic Routing in Intermittently Connected Networks, in Lecture Notes in Computer Science, 2004, pp. 239-254.

[17] J. Burgess, B. Gallagher, D. Jensen, and B. N. Levine. MaxProp: Routing for Vehicle-Based Disruption-Tolerant Networks. *In Proc. IEEE INFOCOM,* 2006.

[18] P. Hui, J. Crowcroft, and E. Yoneki. Bubble-Rap: Social-Based Forwarding in Delay Tolerant Networks. *In Proc. of MobiHoc*, pp. 241–250, 2008.

[19] R. Shah, S. Roy, S. Jain, and W. Brunette. Data MULEs: Modeling a Three-tier Architecture for Sparse Sensor Networks. *in IEEE SNPA Workshop*, 2003.

[20] IEEE Standards Department. Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, *IEEE standard 802.11–1997*.

[21] J. Broch, D. Maltz, D. Johnson, Y-C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. *In Proceedings of IEEE/ACMMOBICOM’98*, pp. 85–97, 1998.

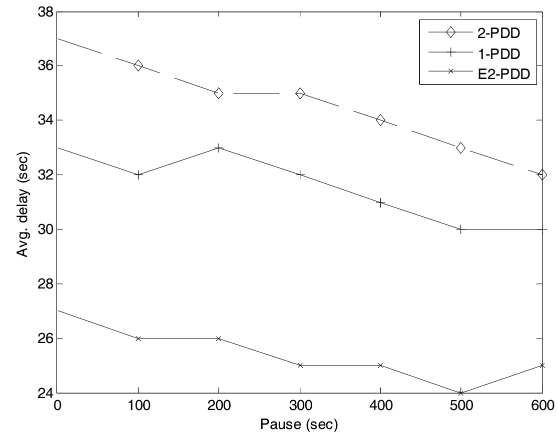


Fig. 5. Avg. delay vs. pause time (with 25 consumers).

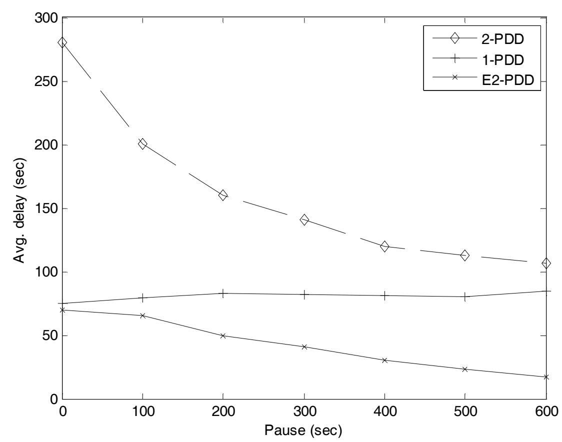


Fig. 6. Avg. delay vs. pause time (with 250 consumers).

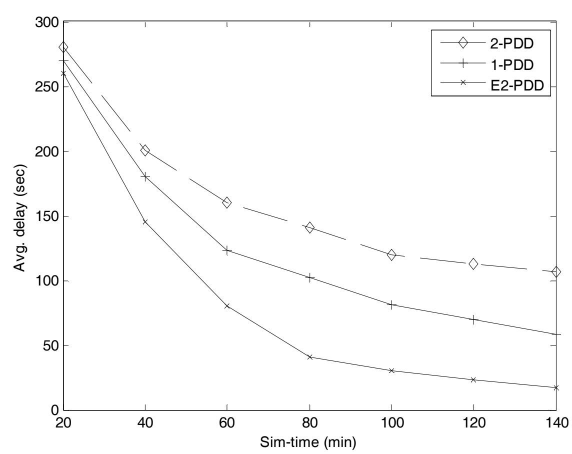


Fig. 7. Avg. delay vs. simulation time (with 250 consumer).