A Probabilistic Interest Forwarding Protocol for Named Data Delay Tolerant Networks

Paulo Duarte, Joaquim Macedo, Antonio Duarte Costa,
Maria João Nicolau, and Alexandre Santos

Centro ALGORITMI, Universidade do Minho, Portugal
{a58655@alunos,macedo@di,costa@di,joao@dsi,alex@di}.uminho.pt

Abstract. Delay Tolerant Networks (DTN) were designed to allow delayed communications in mobile wireless scenarios where direct end-to-end connectivity is not possible. Nodes store and carry packets, deciding whether to forward them or not on each opportunistic contact they eventually establish in the near future. Recently, Named Data Networking (NDN) have emerged as a completely new paradigm for future networks. Instead of being treated as source or destination identifiers, nodes are viewed as consumers that express interests on information or producers that provide information. Current research is carried on the combination of these two concepts, by applying data-centric approach in DTN scenarios. In this paper, a new routing protocol called PIFP (Probabilistic Interest Forwarding Protocol) is proposed, that explores the frequency of opportunistic contacts, not between the nodes themselves, but between the nodes and the information, in order to compute a delivery probability for interest and data packets in a Named Data Delay Tolerant network (ND-DTN) scenario. The protocol design and a prototype implementation for The ONE Simulator are both described. Simulation results show that PIFP presents significant improvements in terms of interest satisfaction, average delay and total cost, when compared to other ND-DTN approaches recently proposed.

1 Introduction

Delay Tolerant Networks (DTN) [3] emerged to deal with connectivity disruption in many scenarios, for instance interplanetary communications [1], military ad-hoc networks, remote places with no communication infrastructures or even in urban scenarios where portable devices carried by humans can communicate between them spontaneously. At a given time, a node may not have a path to a destination node, either because obstacles obstructed the communication or due to nodes mobility. The opportunity to a new contact may be expected (as in space orbits or public transportations) or unknown (as in human movements). All nodes must behave as routers, storing the packets and waiting for a chance to forward them to another node, either the destination node or a good intermediary candidate. For that reason, this model of communication is sometimes called by store-carry-and-forward in opposition to traditional store-and-forward model used in direct end-to-end communications.

The most challenging problem in DTN is how to route packets [13]. An epidemic approach may be used, forwarding packets on every opportunistic contact, with good

N. Mitton et al. (Eds.): AdHocNets 2015, LNICST 155, pp. 94–107, 2015.
DOI: 10.1007/978-3-319-25067-0_8

results in terms of delivery probability but high global network overhead. Other approaches include direct delivery, where a packet is delivered directly by the source to destination. Probabilistic approaches, based on contact history statistics, have also been experimented. One example is PROPHET (Probabilistic Routing Protocol using History of Encounters and Transitivity) [6] that computes a delivery probability $DP = P(a,b)$ for every node a and every destination b. Nodes with higher delivery probability to a destination are better forwarders. At each opportunistic contact, the two nodes exchange between them the DP vectors, and update their own information. Then, based on the final values of DP vector, the forwarding decision is taken. PROPHET provides great improvements in packet delivery ratios when compared to epidemic protocol.

Named Data Networking (NDN) [12] is a new architecture designed to meet current and future needs of the Internet. NDN shifts communication paradigm from host-centric to data-centric. This new paradigm differs from the current Internet in some respects. First, all contents are identified following a naming scheme based on URIs. Second, the communication is driven by the consumer. When a user needs data, it shows interest in it by sending an interest packet. When an interest packet reaches a node that has the desired content, a data packet is sent back. Third, storing packets on the network facilitates content delivery. NDN provides native support for mobility since there is no association between the identification and the location of information. The main components of each network node in this architecture are Forwarding Information Base (FIB), Pending Interest Table (PIT), and Content Store (CS) as shown in Figure 1.

NDN and DTN architectures have been developed for different purposes. DTN architecture is dependent on a model based on communication entities, unlike NDN architecture that focuses on data, enabling storage and reuse of data on the network. Despite of different purposes, these architectures have some similarities: flexible routing, data transport preventing loops and network packet storage. The last item is used by DTN for persistence and interruption tolerance and by NDNs to reduce latency, interruption tolerance, increase reliability and achieve higher performance.

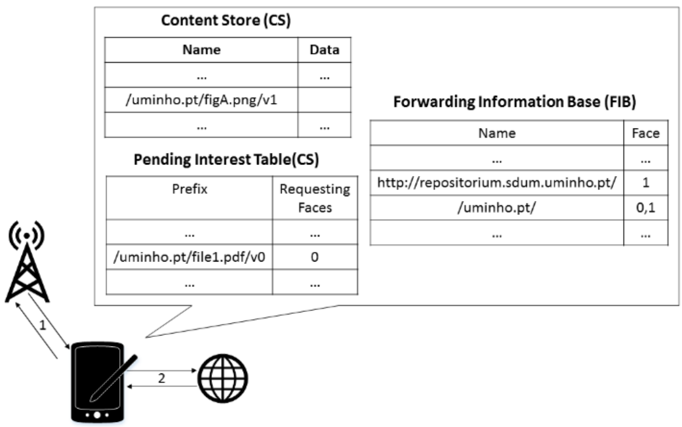


Fig. 1. NDN Architecture based on [9]

In this paper we explore new data-centric forwarding strategies for delay tolerant network scenarios. The proposed protocol, called PIFP (Probabilistic Interest Forwarding Protocol) is based on delivery probabilities computed based on statistics updated on node contacts and is inspired on a similar approach followed by PROPHET. The rest of the paper is structured as follows. Section 2 presents brief overview of relevant related work. Section 3 discusses protocol design issues in detail. Section 4 details the implementation efforts in *The ONE Simulator* [5]. Section 5 presents and discusses the results obtained in simulation. Finally the conclusions and future work in Section 6

2 Related Work

Recently, several proposals have emerged with the aim of exploring the potential of combining the concepts of NDN and DTN architectures to deal with the existing problems in DTN networks.

NAIF (Neighborhood-Aware Interest Forwarding) [11] is a routing protocol that uses named data adapted to MANETs. This approach emerges as an improvement to the original NDNF protocol [4], also based on named data. Unlike NDNF, which broadcasts interests, NAIF nodes cooperatively propagate interest packets from consumer to data sources based on forwarding statistics. A relay node decides to broadcast or drop interest packets based on data packet transmission statistics in its neighborhood.

CEDO (Content-Centric Dissemination Algorithm) [2] is an algorithm for dissemination of content in delay tolerant networks. The aim is to maximize the delivery rate in a distributed environment, where contents that can be ordered and stored have different degrees of popularity. By defining a *delivery-rate utility per content*, this protocol make appropriate decisions on scheduling and content management.

The STCR (Social-Tie based Content Retrieval) [7] is an algorithm that allows retrieval of content in Named Data Delay Tolerant Networks. Over time, nodes store information that will be used later to make forwarding decisions. Using K-mean clustering algorithm, a hierarchical graph is computed based on social ties between nodes. When there is no direct contact with the producer of the content, each node transmits the interest packet only to a node belonging to a higher hierarchy set, with higher social level.

3 PIFP Protocol Design

In this Section, we describe PIFP protocol, whose goal is to increase the number of messages delivered in a named-data network where the nodes are able to store, carry and forward messages in order to deal with frequent connection failures and long periods of time without connectivity, typical in high mobility networks.

3.1 Overview

The approach presented in this paper aims to identify the most advantageous intermediate node to forward an interest packet and it is based on the following main principles:

- The nodes presents mobility patterns and based on these patterns it is possible to predict new encounters between nodes and their content;
- A higher frequency of contact with a specific content indicates that there is a strong likelihood of a new contact with that content;
- Forwarding interests and contents based on names and not on addresses enables better strategies to data storage and reuse.

Inspired on PROPHET, PIFP uses frequency and freshness to evaluate if nodes have a strong relationship with a specific content, but in PIFP, information about the contacts between a node and a specific content is used, instead of the contacts between nodes. This is because PIFP is focused on information and not on hosts. It is assumed that if a node lies often with a content, it is very likely that it will meet again with that content in the near future.

The frequency represents how many times one node meet with one specific content. The freshness reflects how much this information is updated. These two parameters are combined in one metric, called Delivery Predictability (*DP*), which is maintained for each interest in the FIB of a node. So, every node maintains a set of Delivery Predictabilities (*DP*) on the FIB, that are updated on each encounter. The Delivery Predictability $DP(a,c)$, indicates the probability of node a to encounter again the content c.

When a node detects another node in the neighborhood, a connection is established (encounter) and the node records the timestamp of the event. After that, it uses the decay Equation 1 where $\gamma = 0.98$ is the aging constant, and κ is the number of time units (seconds) that have elapsed since the last time the metric was aged. This equation is applied to all content *DP* values that exist in the FIB of the node. *DP* value is affected due to its age. It will decrease over time.

$$DP(a,C_i) = DP(a,C_i)_{old} \times \gamma^{\kappa} \quad (1)$$

After this, two types of information are exchanged between the nodes: FIB routing information and cached data summary. Then Equation 2 is used to update the *DP* value of each content within cached data summary of the other node. The idea is to increase $DP(a,C_i)$ whenever the node a meets C_i. In the first contact with the content, $DP(a,C_i)_{old}$ is 0. The variable α is a scaling factor that sets the rate at which the Delivery Predictability increases in encounters.

$$DP(a,C_i) = DP(a,C_i)_{old} + [1 - DP(a,C_i)_{old}] \times \alpha \quad (2)$$

Through information that is exchanged by FIBs, the predictability of each node to find the contents can be updated. To do so, we can use the PROPHET transitivity property witch was adapted to handle contents. The original rule of transitivity used by PROPHET says that if a node a frequently encounters node b, and node b frequently encounters node c, then node b probably is a good node to forward packets from a destined for node c. To deal with the data-centric paradigm, this rule was rewritten in another way: in a connection between a pair of nodes a and b, if b is often in contact with content C_i, then node b will probably be a good node to forward interests belonging to node a for content C_i.

In Equation 3 we can see the result of the rewritten rule, where $DP(a,C_i)$ is the predictability of a finding C_i and $DP(b,C_i)$ is the predictability of b finding C_i. β is a scale factor that adjusts the weight of the transitive property. If β is defined as zero, the transitive property has no effect and Delivery Predictability is based only in the direct encounters with the contents. High values for β, increases the impact of the transitive property in the likelihood of delivery. In the implementation of PIFP the value set for α variable was 0.75, and value set for β was 0.25. Thus, the property of transitivity has a smaller impact when compared to the use of direct encounters with the contents. Nodes with greater delivery predictability will be those that were in direct contact with the contents.

$$DP(a,C_i) = DP(a,C_i)_{old} + [1 - DP(a,C_i)_{old}] \times DP(b,C_i) \times \beta \tag{3}$$

The main goal of using Delivery Predictability metric is to reduce the number of interest messages sent and to improve the message delivery ratio. This is accomplished because interest messages will only be transmitted to the nodes with higher delivery predictability for that content.

3.2 Information Exchange

As mentioned earlier, each node maintains a CS and FIB table. Data packets passing through the node are temporarily kept in the CS. FIB maps a content identifier with the predictability of an encounter with the same content. During the contact period, the nodes announces its cached content name digest and a summary of its routing table with the corresponding probabilities.

Pseudocode 1. Convergence Process

for all connections to another node **do**
    apply decay equation (1) $[DP(a,C_i) = DP(a,C_i)_{old} \times \gamma^K]$ to own DP sets
    for all cached-content digest received from connected node b **do**
        if there is NO match with local CS **then**
            apply equation (2) $[DP(a,C_i) = DP(a,C_i)_{old} + [1 - DP(a,C_i)_{old}] \times \alpha]$
            update the result into FIB
    for all routing digests received from connected node b **do**
        check my local CS and content digest from other node
        if there is NO match with local CS **then**
            apply equation (3) $[DP(a,C_i) = DP(a,C_i)_{old} + [1 - DP(a,C_i)_{old}] \times DP(b,C_i) \times \beta]$
            update the result into FIB

At each encounter, DP values are updated as shown in Pseudocode 1. When a node receives a summary of content that the other node has cached (direct contact with data), the corresponding DP values are increased by using the Equation 2. When a node receives the routing digest from another node (transitive contact), it will also increase the DP values by applying Equation 3, but only to contents not stored in the two nodes. The calculated values are saved in FIB table for further forwarding of messages.

The process of convergence is very slow due to the long delays caused by DTNs. Over time the nodes will exchange routing information and eventually, the node will have a high knowledge of contents on the network.

3.3 Forwarding Strategy

As described above, the communication is driven by the consumers showing interest in content, i.e., when a node wants to receive data it shows interest on this data by sending one message of interest. The forwarding of interests messages is achieved based on the probability of other node to meet the content. Each node carries interest packets and forward them to a node with a higher Delivery Predictability (*DP*) for that content than itself.

Despite the existing delay in the convergence process, the nodes will have an extended knowledge about the existing contents on the network, which tends to increase the number of extra interest messages sent over the network. To reduce this number, the last *DP* for which the interest was sent is recorded in the PIT ($LastDP(C_i)$). Thus, the interest is sent only if the $DP(C_i)$ of the other node is higher than the $LastDP(C_i)$.

Pseudocode 2. Processing pending interest messages

```
for all connected node b do
    for all pending interest message in node a PIT do
        if DP(b,C_i) > DP(a,C_i) ∧ DP(b,C_i) > LastDP(C_i) then
            add interest message in C_i to the outgoing list O
    if outgoing list O is not empty then
        order list by predictability value
        update LastDP and sends messages to the other node
```

The strategy of sending out interests showed in Pseudocode 2 is not applied when the other node has cached content for the respective interests. When a node receives the summary of data cached, it checks whether there are data that can satisfy the pending interests. If so, the respective messages of interest are sent.

Whenever a node receives an interest packet, it checks the CS table for contents corresponding to this interest. In case of content matching in the CS, the content is delivered to the other node. Otherwise, the unsatisfied interest is stored in PIT, but only if node has knowledge of the content (routing entry in FIB). Only one entry is created on the PIT for the same interest. The identifier of the node that sent the interest is also stored.

Pseudocode 3. Processing incoming interest message

```
for all interest packet on C_i received do
    check my local CS
    if there is a match then
        send the content message to the other node
    else
        check my local FIB
        if there is a match then
            add interests to the PIT
```

PIT also stores information about the validity of its inputs, which are discarded after expired. A Time-To-Live (TTL) is configured in interest packet and decreases over time. A routing cycle will causes the TTL to reach zero, removing the PIT entry. To relax the validity of entries in the PIT, when an existing interest packet is received, the TTL is checked and the validity of the input in the PIT is updated, by incrementing the difference between the TTL of the arrived interest packet and the current TTL of the PIT entry.

After the interest reaches the content producer, a content message is created and sent back trying to reach the consumer. Over time, interest messages are forwarded and "crumbs" are left on the network for subsequent forwarding the content message to consumers. Multiple paths are created, increasing the likelihood of reaching the consumers.

Finally, when a content packet reaches a node, the PIT table is checked for interests corresponding to it, as shown in Pseudo-code 4. If correspondence does not exist in PIT, data is simply discarded. To ensure free storage space, a life time to the contents stored in the CS is defined. In this way, the older contents are eliminated leading to contents that were requested most recently.

Pseudocode 4. PIFP: Processing incoming content message

for all incoming content message C_i received **do**
    check my local CS and my buffer
    if there is a match **then**
        discard the content C_i
    else
        check my local PIT
        if there is a match **then**
            store the content C_i
            if isSubscriber() **then**
                set satisfied interest
        else
            discard content C_i

Assuming the nodes have storage space available they could store the contents even if there were no correspondence in PIT. This approach is exploited by creating a variant of PIFP, called *PIFP-Proactive*, with proactive transmission and storage. This mechanism may improve performance in delivering the contents, but will increase the network load. In this variant, the mechanism for content distribution has been modified, by removing the restriction of transferring data for the same node which has received the interest. It was also removed the verification of valid PIT entries to enable the creation of new entries in CS. Thus, any content received by the node can be stored unless there is no storage space as shown in pseudo-code 5. This approach provides a proactive content delivery, spreading several replicas of the message content with the goal of reaching the node that requested the message content.

Pseudocode 5. PIPF-Proactive: Processing incoming content message

```
for all incoming contente message of C_i received do
    check my local CS and my buffer
    if there is a match then
        discard the content C_i
    else
        check my local PIT
        if there is a match then
            store the content C_i
            if isSubscriber() then
                set satisfied interest
        else if (there is enough space in my CS) then
            store the content C_i
```

4 Implementation

PIFP was implemented on a specially modified version of *The ONE* (Opportunistic Network Environment simulator [5], v1.5.1 RC2), called ICONE (Information Centric ONE [8], v1.0).

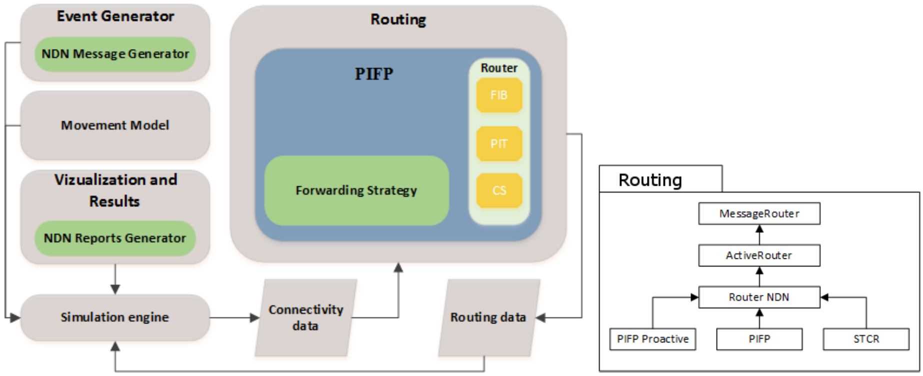
The ONE is a well known DTN simulator, written in Java, originally designed for accessing the performance of DTN protocols. The approach followed was to provide only a simplified model of the lower layers (physical and logical), giving more importance to applications, routing protocols and good mobility models, either synthetic ones or based on real life traces. Settings can be adjusted to create distinct scenarios, changing the frequency, duration and other properties of opportunistic node encounters.

However, The ONE supports only a node-to-node communication model, with source and destination addresses, and has no support for the NDN paradigm. The first step was therefore to extend and modify it, in order to implement a simplified NDN model.

4.1 ICONE Overview

Figure 2 shows the architecture of ICONE with the new and modified components. Messages have to carry information names and nodes must incorporate the three basic components of NDN architecture: CS, PIT, and FIB. Forwarding decisions are based on content name matching (best match) in FIB. *NDNRouter* is a major class in this architecture. It extends *ActiveRouter* object in order to incorporate CS, PIT and FIB storage components. All specific routing strategies already implemented (*PIPF*, *PIPFProactive* and *STCR*) are subclasses of *ActiveRouter*, inheriting all basic operations on CS, PIT and FIB structures.

Entries in data structures are stored by name. Names are strings of variable and unbound length. In order to allow faster searches and also an efficient usage of memory, Counting Bloom Filters [9] were used to implement PIT, FIB and CS structures. A bloom filter consists of a bit vector of length m, initialized with all bits set to zero. Whenever a new element is added, multiple hash functions are used to compute k hash positions. The correspondent bits in those positions are set to 1. The exact same k bits



(a) Extensions and Modifications for NDN support **(b)** NDN Routing Classes

Fig. 2. ICONE Framework

are verified on a element lookup operation. If they are all set to 1, there is a match with limited probability of being a ”false positive”. Otherwise, there is no match, with zero probability of ”false negatives”.

4.2 ICONE Components

A new *NDNMessage* was defined with a common set of fields. Instead of the usual ”to” and ”from” fields, useless in NDN networks, a *NDNMessage* contains the *ContentName*, *MessageType* and *TTL*. All messages have a finite TTL and are discarded when that TTL expires. Two types of messages were defined: *Interest* and *Data*. Consumers express their ”interest” on a specific content by creating a message with type *Interest*, entering the name of content into *ContentName* field and sending it to the network. When an *Interest* message reaches a node with the content in the CS (either a producer or a caching node) a reply message of type *Data* is sent back.

Message Generator. In this paradigm, all communications are driven by consumer through the transmission of interest packets. Distinct application scenarios can therefore be created by the way consumers generate interests on data. A message generator module was created in order to generate events. Each event corresponds to the generation of an interest message by a given node. All events are generated at once, according to a set of input parameters, and stored in a trace file. During simulation, the trace file is loaded and events added to the scheduler on the right node.

Current implementation tries to mimic realistic web browsing sessions, based on the study reported in [10]. All important features were identified and are preserved by the message generator, like the number of requests per session, time between sessions and request size. In addition, data message events are also generated when a producer or intermediate caching node receives an interest whose name matches an entry. The size of data can be configured between a minimum and a maximum.

Content Store. Producers are initialized with a set of contents. The CS is populated with the contents. When an interest arrives, CS is checked to see if there is a match.

Network Faces. Unlike static networks, within a mobile ad-hoc wireless network there is no concept of face. In ICONE, the IDs of node are used as a Face. When a packet of interest arrives, the associated incoming face is the ID of the requesting node.

NDN Reports. Due to the change of paradigm, a few new reports were implemented. A set of metrics can be obtained during simulation, including the number of interest packets generated, the number of interest packets sent, the number of satisfied interests, the number of data packets sent, the number of satisfied interest in the local node, the delay and message size.

5 Performance Evaluation

PIFP was experimentally evaluated using ICONE and compared *PIFP-Proactive* and *SCTR*. *PIFP-Proactive* is a variant of *PIFP* described in previous section (see pseudo-code 5). STCR was originally designed for delay-tolerant MANETS. It builds a social hierarchy and groups nodes together in clusters using the k-mean algorithm. *SCTR* was also implemented from scratch on ICONE.

To evaluate PIFP, the following metrics were used: *HitRate* (number of interests satisfied on the node and by the network), *AverageDelay* (average delay time, measured from the generation of the interest packet up to the reception of content that satisfies it), and *TotalCost* (total number of interest packets and data packets in the network).

5.1 Simulation Scenario

The proposed solution was evaluated using two different scenarios. Table 1 details the parameters for both scenarios. These two scenarios were selected in order to evaluate the performance of PIFP in dense and sparse networks. As in PROPHET, the PIFP may have a higher rate of delivery in denser networks. The convergence of the names is expected to be faster. In a sparse scenario, due to lack of connectivity, the interest rate is expected to be lower and with higher message delivery delays.

In the first scenario, a more dense opportunistic network is created, with a total of 98 nodes. Nodes consist of two groups of pedestrians (one with thirty two elements and the other with thirty), a group of cars (thirty two elements) and two groups of trains (with two trains each). Nodes move according to the map of Helsinki, Finland. Pedestrian group move according to the Shortest Path Map-Based model. Cars are forced to follow only on the roads using the same movement pattern of pedestrians. Trains use the Routed Map-Based Movement Model. In the second scenario, the total number of nodes was reduced to 49, in order to create a sparser network.

The *MessageGenerator* module was used to create a trace file modeling several user web sessions. Given a simulation time, the generator produces a number of sessions distributed along the entire time of the simulation. Each session has a number of requests. A request in this paradigm represents an interest message. When the simulation starts, the events are loaded from a file. A different message set was generated for each scenario, according to the simulation time and the number of nodes. The module is parameterized with the number of nodes in the network and the simulation time. The larger the value of these parameters, the greater the number of generated interests.

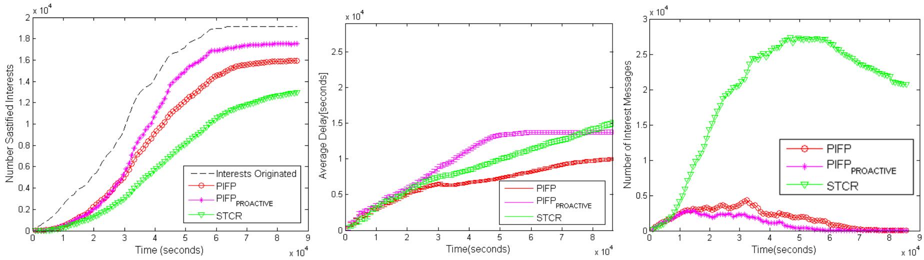
Table 1. Simulation Parameters

Parameter	**Value**
Simulation Time	86400s
Deployment field	4500m×3400m
Node speed	Pedestrian: 0.5 to 1.5 m/s Cars: 2.7 to 13.9 m/s Trams: 10 to 30 m/s
Request lifetime	8 hours
Radio	Bluetooth
Transmission range	Pedestrians and Cars: 10m Trams and Special Pedestrians: 1km
Transmission rate	Simple Interface: 250kbps High Speed Interface: 10Mbps
Storage capacity	Consumers: 50MB Producers: 100MB

5.2 Experimental Results

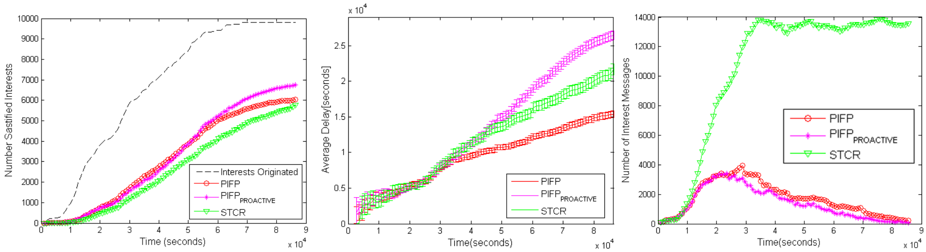
Figures 3 and 4 show the results obtained by the three different strategies (*PIFP*, *PIFP-Proactive* and *SCTR*) in the two simulation scenarios (dense and sparse).

Number of Interests Satisfied. Regarding the *HitRate*, represented in Figure 3a (dense scenario) and Figure 4a (sparse scenario), graphs show that the number of satisfied interests increases with the number of nodes in the network. Since there are more nodes



(a) Interests Satisfied **(b)** AVG Delay **(c)** Total Interests

Fig. 3. Results for Scenario 1 (dense)



(a) Interests Satisfied **(b)** AVG Delay **(c)** Total Interests

Fig. 4. Results for Scenario 2 (sparse)

in the network, there is a greater number of distributed caches. Intermediate nodes, that have contents, become also content servers.

Comparing the three approaches, one can see that the *PIFP-Proactive* scheme has the highest number of satisfied interests in both scenarios. Its proactive delivery of contents creates more opportunities to reach the consumers. Proactive PIFP achieved 92% satisfaction of interests on Scenario 1 (dense). Furthermore, the PIFP overcomes STCR. The interest satisfaction rate for STCR protocol is about 70% while for PIFP is approximately 83%. In Scenario 2 (sparse), the schemes PIFP, STCR and *PIFP-Proactive* had similar behavior. The number of satisfied interests was related.

Figure 5a shows the number of interests satisfied locally by the node or by the network. When a node generates an interest, the CS is checked to see if there is any correspondence. If so, then the interest is locally satisfied. The PIFP presents a higher number of satisfied interests in the network than locally. In Proactive solution approximately 60% of interests are satisfied locally. PIFP and STCR satisfy respectively about 31% and 45% of interest in the node.

Average Delay. Figure 3b and Figure 4b illustrate the average delay over time for the three protocols. Only delays for interests satisfied with data received from the network are presented. Interests satisfied in the node itself are not included.

Once an interest packet arrives to a producer or intermediate node in procession of the content, a data message is sent back to the consumer. In the case of *PIFP-Proactive* the data is sent to any node with available cache space. In case of PIFP data is sent to nodes that have the interest in the PIT. Finally, in the case of STCR, the data is only sent to nodes with a stronger social relationship with the consumer.

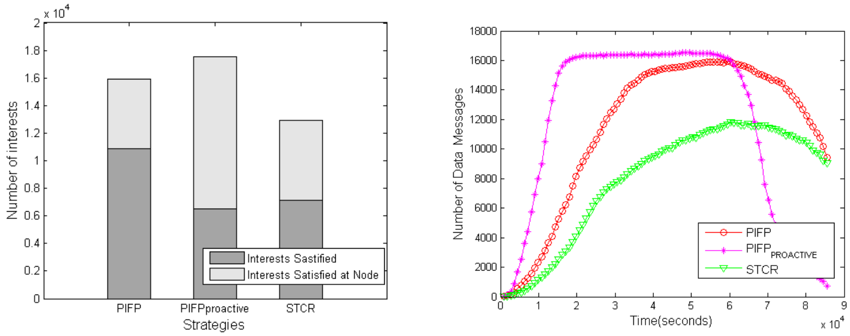
Apparently, the lower average delay time should occur for the strategy that gets more data paths to the consumer, which is *PIFP-Proactive*. However, the network flood with the data sent using this strategy, limits the data storage for subsequent interests. This reason makes the average waiting time for *PIFP-Proactive* greater than the PIFP. PIFP, in turn, obtains more paths than the STCR. The amount of data in the network for the different strategies can be seen in Figure 5b.

At the beginning of simulation, the sending of interest packet by an STCR node is faster since it is made to nodes with greater centrality. Unlike STCR, PIFP sends only interest packet to a node with knowledge of the associated content. Due to the slow convergence of routing tables at beginning of simulation, many contents may be unknown. This justify the better results of STCR at beginning of simulation.

In Figure 4b(Scenario 2), the behavior was similar for all approaches. The delay was increased when compared with Figure 3b (scenario 1). This is due to the fact that the network is not as dense as in Scenario 1. The number of available paths decreases and this had an impact in the delivery of contents.

Total Cost. Regarding *TotalCost*, the number of interests in the network was evaluated in both scenarios and also the number of data packets in scenario 1.

As shown in Figure 3c and Figure 4c, STCR has an higher number of interests in the network, when compared with the remaining protocols. STCR initially transmits the interests to nodes with highest centrality and from different clusters. When the producer



(a) Network vs Node Satisfaction **(b)** Total number of data packets (scenario 1)

Fig. 5. Other results

is known the interest is sent to him using social relationships. Apparently, this approach makes more interest replicas than the one used by PIFP, which sends the interests only to nodes with higher predictability to reach the content. Furthermore, PIFP discards interest messages when the interest is satisfied or has a TTL expired. STCR does not discard the interest message when the TTL expires, and it is sent epidemically, which increases the number of replicas in the network.

Figure 5b presents the total number of data packets for Scenario 1 (dense). As described before, *PIFP-Proactive* sends data packets whenever the neighbors nodes have free space in cache. So this strategy has the higher number of data packets in the network. PIFP sends the data packet to all nodes with the interest in the PIT. STCR send the data packet to nodes with higher social relationship with the consumer.

6 Conclusions

In this paper, a new routing protocol called PIFP (Probabilistic Interest Forwarding Protocol) is presented. PIFP uses a probabilistic approach to forward interest in a Named Data Delay Tolerant Newtork. PIFP is inspired on PROPHET, a well known protocol for Delay Tolerant Networks. Like PROPHET, PIFP computes a delivery probability based on the history of encounters. But while PROPHET records the frequency of encounters between the node and the other nodes, PIFP records the frequency of encounters between the node and the data contents. It is therefore adapted to the named data networking paradigm. A proactive variant of PIFP protocol is also described. The main difference is that proactive version stores data packets, proactively, in all forwarding nodes that have enough free space in the cache memory. The goal is to increase data availability and reduce the delivery time to the consumer, at the expense of available storage space, and without extra communication overhead.

The performance of the new protocols was evaluated in comparison with STCR, an existing NDN forwarding mechanism for MANETs. STCR is based on social relations between devices, while PIFP constructs relationships between the devices and the contents. The three protocols were implemented in a modified version of ONE, called ICONE. ICONE results from extending and modifying ONE in order to support the named data paradigm. For the two simulated scenarios (dense and sparse), results obtained show that the proposed PIFP protocols present better values in terms of the number of interest satisfied, the average delay and total number of interest packets in network. STCR obtains better values on the total number of data packets transmitted in the network. More extensive evaluation, in distinct scenarios, is however required.

As future work, we plan to use this named data based protocol in scenarios with selfish nodes. Knowing in advance the interests of a given node may reduce the uncertainty about his behavior when processing a given packet.

Acknowledgments. This work has been supported by FCT – Fundação para Ciência e Tecnologia, in the scope of the project: UID/CEC/00319/2013.

References

1. Burleigh, S., Hooke, A., Torgerson, L., Fall, K., Cerf, V., Durst, B., Scott, K., Weiss, H.: Delay-tolerant networking: An approach to interplanetary internet. IEEE Commun. Magazine 41(6), 128–136 (2003)
2. De Meneses Neves Ramos Dos Santos, F., Ertl, B., Barakat, C., Spyropoulos, T., Turletti, T.: CEDO: content-centric dissemination slgorithm for delay-tolerant networks. In: Proceedings of ACM/IEEE MSWiM 2013, pp. 377–386. ACM, November 2013
3. Fall, K.: A delay tolerant network architecture for challenged internets. In: Proceedings of SIGCOMM 2003, p. 27. ACM Press (2003)
4. Jacobson, V., Smetters, D.K., Thornton, J.D., Plass, M., Briggs, N., Braynard, R.: Networking named content. Communications of the ACM 55(1), 117 (2012)
5. Keränen, A., Ott, J., Kärkkäinen, T.: The ONE simulator for DTN protocol evaluation. In: Proceedings of SIMUTools 2009. ICST, New York, NY, USA, p. 55 (2009)
6. Lindgren, A., Doria, A., Schelén, O.: Probabilistic routing in intermittently connected networks. ACM SIGMOBILE Mobile Comput. and Commun. Review 7(3), 19 (2003)
7. Lu, Y., Li, X., Yu, Y.T., Gerla, M.: Information-centric delay-tolerant mobile ad-hoc networks. In: Proceedings of IEEE INFOCOM Workshops 2014, pp. 428–433. IEEE (2014)
8. Macedo, J., Costa, A., Nicolau, M.J., Santos, A.: Icone (information centric one) documentation, March 2015. `http://marco.uminho.pt/projects/ICONE/`
9. Quan, W., Xu, C., Guan, J., Zhang, H., Grieco, L.A.: Scalable name lookup with adaptive prefix bloom filter for named data networking. IEEE Commun. Letters 18(1), 102–105 (2014)
10. Schneider, F., Agarwal, S., Alpcan, T., Feldmann, A.: The new web: characterizing AJAX traffic. In: Claypool, M., Uhlig, S. (eds.) PAM 2008. LNCS, vol. 4979, pp. 31–40. Springer, Heidelberg (2008)
11. Yu, Y.T., Dilmaghani, R.B., Calo, S., Sanadidi, M.Y., Gerla, M.: Interest propagation in named data manets. In: Proceedings of ICNC 2013, pp. 1118–1122. IEEE, January 2013
12. Zhang, L., Afanasyev, A., Burke, J., Jacobson, V., Claffy, K., Crowley, P., Papadopoulos, C., Wang, L., Zhang, B.: Named Data Networking. ACM SIGCOMM Comput. Commun. Review 44(3), 66–73 (2014)
13. Zhang, Z.: Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges. IEEE Commun. Surveys & Tutorials 8(1), 24–37 (2006)