A Centrality-Based ACK Forwarding Mechanism for Efficient Routing in Infrastructureless Opportunistic Networks

Sanjay K. Dhurandher[1], Isaac Woungang[2(✉)], Anshu Rajendra[1], Piyush Ghai[1], and Periklis Chatzimisios[3]

[1] CAITFS, Netaji Subhas Institute of Technology, University of Delhi, New Delhi, India
dhurandher@gmail.com, {anshurajendra,piyushghai}@nsitonline.in
[2] Department of Computer Science, Ryerson University, Toronto, Canada
iwoungan@scs.ryerson.ca
[3] Department of Informatics, ATEITHE, Thessaloniki, Greece
peris@it.teithe.gr

Abstract. In the next generation Internet, it is expected that human, smart devices, and "things" will be able to communicate and interact with each other opportunistically in order to share their data. An analysis of this type of relationship is made possible due to the advent of the Opportunistic Internet of Things (OppIoT), a new paradigm that enable information sharing and dissemination among opportunistic communities formed based on human mobility and opportunistic contacts. Designing data dissemination and routing protocols for opportunistic IoT is a challenge since contacts between nodes and users' social behaviors are to be tighten together as design constraints. Nonetheless, as opportunistic networking rely on spontaneous connectivity between the users and wireless devices, it can be argued that OppIoT is a form of opportunistic social networks (OppNets) extension, with focus on the relationship between human and opportunistic connection of smart things. As such, some routing protocols that have been designed to work for infrastructureless OppNets can also be applied in OppIoT systems. In this context, the History Based Routing Protocol for Opportunistic Networks (HiBOp) is an appealing choice. In this paper, an acknowledgement (ACK) forwarding mechanism to boost the performance of HiBOp is proposed based on the concept of centrality. Simulation results are provided, showing that HiBOp with centrality outperforms HiBOp in terms of predefined performance metrics.

Keywords: Opportunistic networks (OppNets) · Centrality · ACK · Routing protocol · HiBOp

1 Introduction

The next generation Internet is expected to provide facilities for human and smart things to connect and interact with each other. Examining the social

B. Mandler et al. (Eds.) IoT 360° 2015, Part I, LNICST 169, pp. 99–108, 2016.
DOI: 10.1007/978-3-319-47063-4_9

side of Internet of Things (IoT) from a human-centric perspective is the goal of OppIoT [1], a novel computing paradigm and technology that promotes the idea that smart 'things' can be opportunistically connected with human user by means of short-range communication and sensing technologies. Typically, an OppIoT system can be viewed as an ad hoc infrastructureless OppNet architecture of devices that can enable data sharing and dissemination within opportunistic communities formed based on human mobility and opportunistic contacts. In [1], a reference architecture for the development of an OppIoT system is proposed. In contrast, in infrastructureless oppNets [2,3], an instantaneous connectivity is established among the users equipped with wireless devices, forming a kind of decentralized ad hoc network that allows inter-device data routing and forwarding. The sender of a message has no proof of knowledge about its delivery and node mobility is exploited to send the messages in a disconnected environment based on opportunistic communication. In such environment, the transmission and reception of ACK messages remains an issue that has not been addressed by existing routing schemes. Indeed, OppNets make no prior assumption about the network topology and evaluate the delivery probability of a node based on a forwarding strategy, taking into account the limited time in which nodes are in the radio range of each other. Typically, each node has a limited buffer storage; hence, messages drop, node failure, or incomplete transmission may occur, if this buffer is not properly managed.

Unlike conventional flooding techniques, HiBOp [3] makes use of a context-based knowledge to calculate the delivery probability from a sender to a receiver, but the ACK management and buffer saturation are not considered in its design. In this paper, a centrality-based mechanism for the delivery of ACKs from destination to source is implemented on top of HiBOp to help clearing the buffers of residual messages, leading to an enhancement in message delivery.

The rest of the paper is organized as follows. In Sect. 2, some routing protocols for OppNets are discussed. Section 3 describes our proposed ACK forwarding mechanism. Section 4 presents some simulation results. In Sect. 5, we conclude the paper.

2 Related Work

Several routing protocols have been proposed for OppNets [4–8], most of which are based on the principle of deciding which messages are to be forwarded or dropped when a peer comes in contact with another as the buffer reaches its maximum capacity. The Epidemic protocol [4] uses a flooding technique with no limit on the buffer capacity. The Spray and Focus protocol [5] utilizes a form of controlled flooding technique to send the messages. The Prophet+ [6], MaxProp [7], HBPR [8] and HiBOp [3] protocols all exploit some form of context-based information to calculate the node's delivery probability. The data processing overhead in these protocols calls for an efficient ACK mechanism to prevent buffer saturation. The HiBOp protocol [3] relies on the following data structures: Identity Table - used by nodes to learn about the context in which they are currently immersed in (e.g. home and work locations, name, etc.); Current Context Table - this stores all Identity Tables of neighbors of a node; History Table - this keeps track of the past information seen by the current node; and Repository Table - used for updating the History Table. Based on these data structures, the delivery probability of a node is estimated. Accordingly, a limited number of neighbors of a node are retained as best candidates for message forwarding purpose.

3 Proposed Centrality-Based ACK Forwarding Mechanism

In HiBOp [3], no ACK mechanism is provided to confirm to the source that a message has been successful delivered. This has motivated the following changes to the design of HiBOp: a list of ACK message IDs is maintained at each node. Whenever two nodes meet each other, the Current Context is updated, and the exchange of ACKs takes place as described in Algorithm 2. Nodes whose ACKs have been received by the source thus relieve their respective buffers. On the other hand, those ACKs which have not yet reached their respective senders (after a timeout period) are forwarded to the best nodes filtered according to the centrality parameter. This parameter exploits the fact that central nodes in the network have the maximum connectivity and will more likely be part of the routing path that is being build on the fly. The home and work locations are used to calculate the centrality of a node as described in Algorithm 1. This algorithm checks whether the node's location has changed or not. If the node is mobile, the previous value of the centrality is retained. Else, the centrality values at home and at work are calculated.

Algorithm 1. Calculation of the centrality of a node.

```
Fetch the location of the node
if Location = Old Location then
    Centrality ← OldCentrality
else if Location = Home Location then
    Calculate centrality C_h at home
    Centrality ← C_h
else if Location = Work Location then
    Calculate centrality C_w at work
    Centrality ← C_w
end if
Centrality ← Max(C_h, C_w)
return
```

Forwarding the ACK message takes into account the centrality with respect to the message. Using Algorithm 1, the most central node on the path followed by the message being transmitted from source to destination is identified and appended to the message. Its *centrality value* is updated on each successive hop of the message subject to some conditions. While sending an ACK, the destination first sends it to the node with maximum centrality on the path taken by the message (as per Algorithm 2). If the destination is not in direct contact with this central node, it forwards the ACK to its most central neighbor (i.e. node with the highest centrality value) for further forwarding. This process continues recursively until the ACK reaches within one-hop distance of the source. Upon receipt of this ACK, the source then clears its buffers of residual messages. Note that during this process, no copy of the ACK is kept within the current node.

Algorithm 2. Forwarding and Exchange of ACK Messages

```
for all message m in Buffer do
  CentralNode ← CentralNode of m
  if CurrentNode = CentralNode then
    Determine the next Central Node id and append it to m
    if OtherNode = CentralNode then
      Forward the ACK and clear the buffer
      if OtherCentrality ≥ CurrentNodeCentrality then
        Forward the ACK and clear the buffer
      end if
    end if
  end if
end for
return
```

4 Simulation Results

The ONE simulator [9] is used, along with the Working Day movement model and a network consisting of 8 groups of 50 nodes each. Each node is initially at its home (or work) location, then travels towards its work (or home) location at a random speed chosen in the range 7–10 m/s after an initial rest period. For generating some randomness in the movement of the nodes, a shopping probability of 0.5 is assumed. This process continues throughout the simulations, causing some dynamic changes in the network topology. It is also assumed that the interpersonal communication between mobile users holds using mobile phones or Bluetooth devices at 2 Mbits/s data rate, with 10 m radio range. Messages are generated randomly by the nodes every 25–35 s. Other main simulation are captured in Tables 1 and 2.

The studied protocols are compared in terms of message throughput over different simulation time intervals. The results are captured in Fig. 1. It is observed that HiBOp with centrality outperforms HiBOp with respect to that metric. This is attributed to the efficient routing of the ACKs that occurs when HiBOp with centrality is used, leading to a reduced buffer occupancy by messages.

Next, HiBOp with centrality is compared against HiBOp in terms of Message Loss when the TTL, buffer size, and Working Day Length are varied respectively.

Table 1. Simulation parameters

Parameter	Value
Signaling interval	5 s
Death interval	10 s
Repository flushing interval	1800 s
Flushing Interval	10 s
Default message size	500 KB–1 MB
Default buffer size	50 M

Table 2. Mobility parameters

Parameter	Value
Message TTL	300 min
Number of offices	50
Default work day length	15 h
Repository flushing interval	1800 s
Shopping probability	0.5
Office size	10
Minimum office wait time	10 s
Maximum office wait time	10000 s

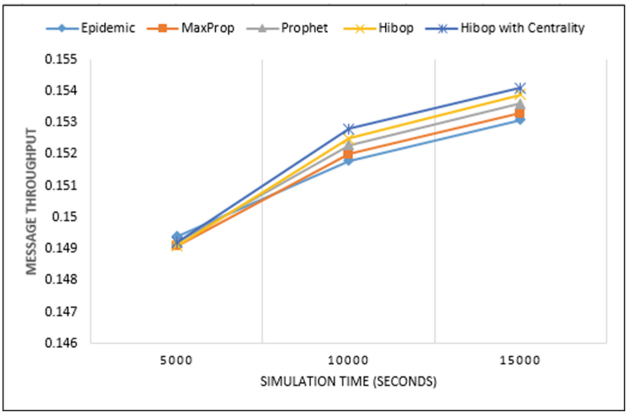

Fig. 1. Message throughput under varying simulation times

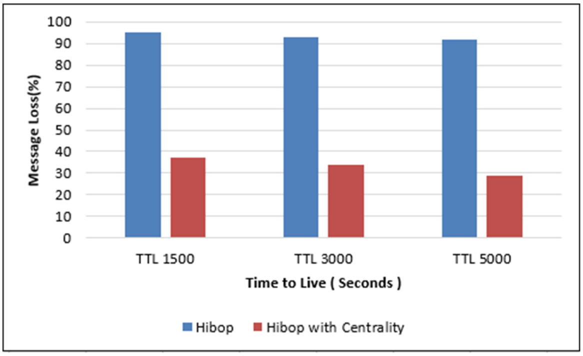

Fig. 2. Message loss under varying TTL

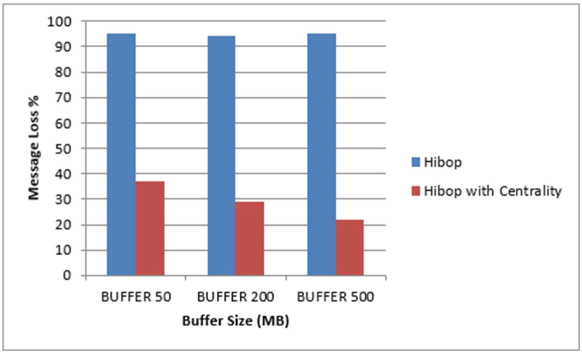

Fig. 3. Message loss under varying buffer size

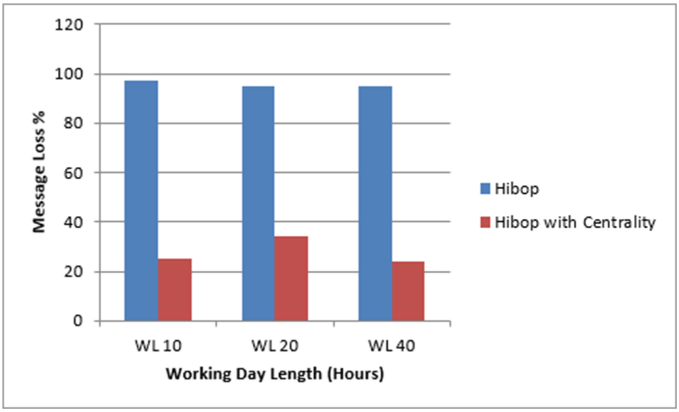

Fig. 4. Message loss under varying Working Day Length

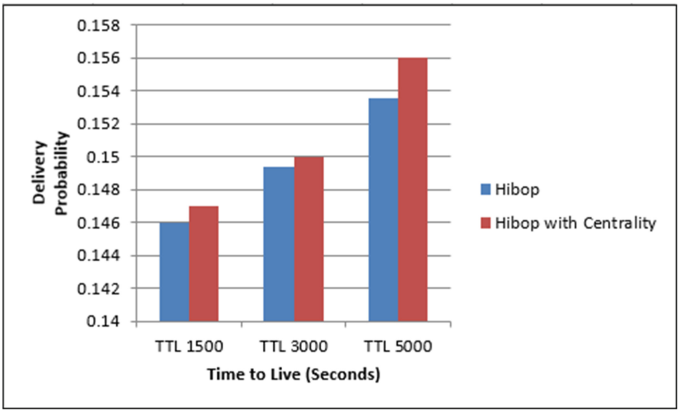

Fig. 5. Delivery probability over varying TTL

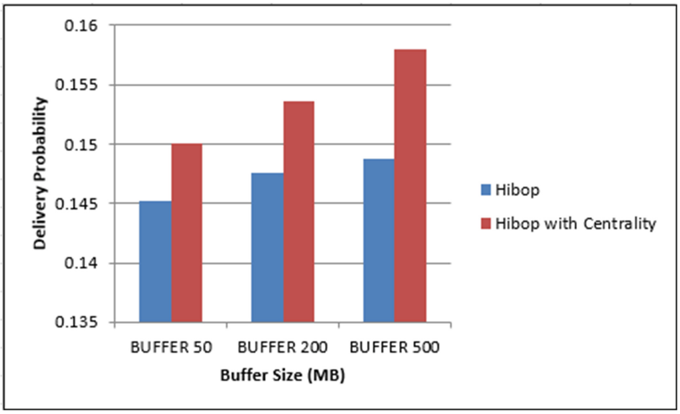

Fig. 6. Delivery probability over varying buffer size

The results are captured in Figs. 2, 3, and 4 respectively, where the Message Loss is obtained:

$$MessageLoss = \frac{Number\ of\ messages\ dropped}{Number\ of\ messages\ generated}$$

In Fig. 2, it is observed that under varying TTL, there is a significant reduction in the Message Loss (about 75–80 %) when HiBOp with centrality is used. Similar results prevail when the buffer size and Working Day Length are varied respectively (as shown in Figs. 3 and 4). When the buffer size and TTL increases respectively, it is observed that the number of Message Loss decreases.

Next, HiBOp with centrality is compared against HiBOp in terms of delivery probability when the TTL and buffer size are varied respectively. The results are

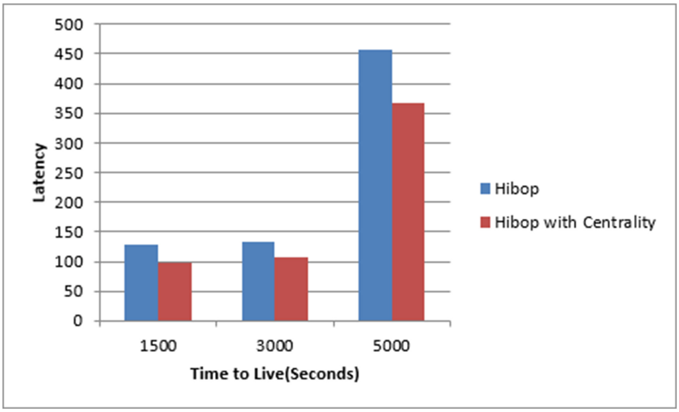

Fig. 7. Latency over varying TTL

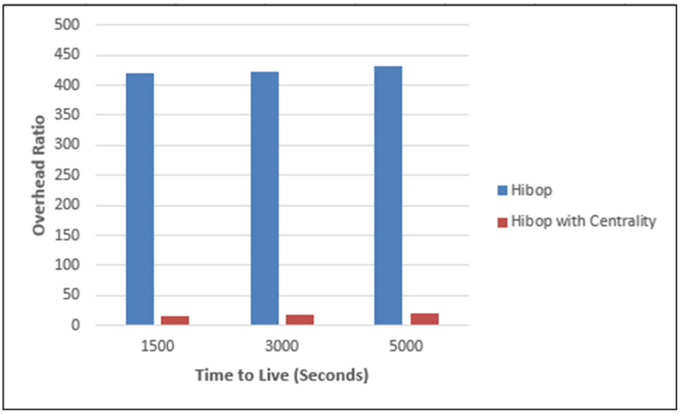

Fig. 8. Overhead ratio over varying TTL

captured in Figs. 5 and 6. In Fig. 5, it is observed that HiBOp with centrality performs better than HiBOp, even with the space constraints introduced by the buffer size. The improvement is in the order of about 5–10 %. This is attributed to the low message loss generated by HiBOp with centrality (compared to that obtained from HiBOp). In Fig. 7, it is observed that HiBOp with centrality yields an improved latency compared to HiBOp (about 30 % improvement). In Fig. 8, it is observed that there is a significant improvement in the buffer overhead generated by HiBOp with centrality compared to that generated by HiBOp (about 90–95 % improvement).

Finally, MaxProp is compared against HiBOp with centrality in terms of buffer occupancy, traffic overhead, and latency. The results are captured in Fig. 9. In Fig. 9, it can be observed that HiBOp with centrality yields an improvement

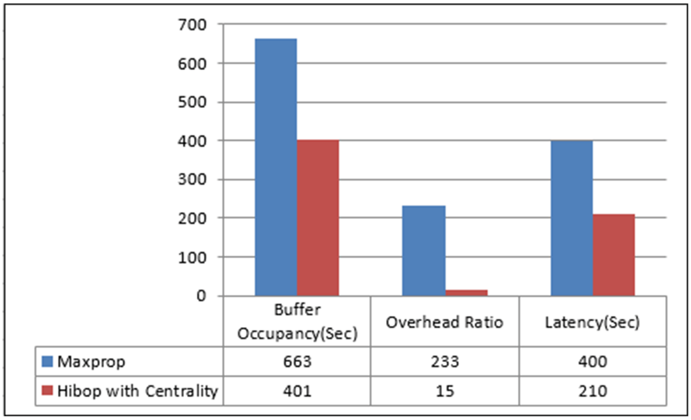

Fig. 9. HiBOp with centrality vs. MaxProp

over HiBOp of about 40 % in terms of buffer occupancy, 50 % in terms of latency, and 90 % in terms of traffic overhead. The above results illustrate the advantages of using our proposed ACK mechanism as a way to reinforce the routing in OppNets.

5 Conclusion

In this paper, a novel approach for the routing of ACKs based on the concept of centrality has been proposed, leading to an enhancement of the HiBOp routing protocol for OppNets. Due to its generic nature, the proposed centrality-based ACK mechanism can be implemented on top of any known routing protocols for OppNets. Simulation results have shown that: (1) this mechanism can significantly reduce the message loss rate while preserving the performance of HiBOp; (2) HiBOp with centrality outperforms the MaxProp protocol in terms of buffer occupancy, traffic overhead, and latency. As future work, the proposed centrality-based ACK mechanism can further be assessed by considering other realistic movement models. In addition, the calculation of the centrality of a node assumes a direct relation to the number of active connections of a node. Other factors such as the duration of active connections can be considered in this calculation.

References

1. Guo, B., Zhang, D., Wang, Z., Yu, Z., Zhou, X.: Opportunistic IoT: exploring the harmonious interaction between human and the Internet of Things. J. Netw. Comput. Appl. **36**(6), 1531–1539 (2013)
2. Boldrini, C., Conti, M., Delmastro, F., Passarella, A.: Context and social-aware middleware for opportunistic networks. J. Netw. Comput. Appl. **33**(5), 525541 (2010)

3. Boldrini, C., Conti, M., Jacopini, J., Passarella, A.: HiBOp: a history based routing protocol for opportunistic networks. In: Proceedings of IEEE International Symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM 2007), Helsinki, 18–21 June 2007, pp. 1–12 (2007)
4. Vahdat, A., Becker, D.: Epidemic routing for partially-connected ad hoc networks, Technical report CS-2000-06. Duke University, July 2000
5. Thrasyvoulos, S., Psounis, K., Raghavendra, C.S.: Spray and focus: efficient mobility-assisted routing for heterogeneous and correlated mobility. In: Proceedings of Pervasive Computing and Communications Workshops (PerCom 2007), White Plains, 19–23 March 2007, pp. 79–85 (2007)
6. Huang, T.-K., Lee, C.-K., Chen, L.-J.: PRoPHET+: an adaptive PRoPHET-based routing protocol for opportunistic network. In: Proceedings of the 24th IEEE AINA, Perth, 20–23 April 2010, pp. 112–119 (2010)
7. Burgess, J., Gallagher, B., Jensen, D., Levine, B.N.: MaxProp: routing for vehicle-based disruption-tolerant networks. In: Proceedings of INFOCOM 2006, Barcelona, 23–29 April 2006, pp. 1–11 (2006)
8. Dhurandher, S.K., Sharma, D.K., Woungang, I., Bhat, S: HBPR: history based prediction for routing in infrastructure-less opportunistic networks. In: Proceedings of the AINA 2013, Barcelona, pp. 931–936 (2013)
9. Keränen A., Ott J., Kärkkäinen T.: The ONE simulator for DTN protocol evaluation. In: Proceedings of the 2nd International Conference on Simulation Tools and Techniques (SIMUTools 2009), Rome, 2–6 March 2009