Content Synchronization with Feedback in Smart City Device-to-Device Communication

Xiao Chen
Department of Computer Science
Texas State University
San Marcos, TX 78666
Email: xc10@txstate.edu

Wenzhong Li
State Key Laboratory for Novel Software Technology
Nanjing University
Nanjing, China
Email: lwz@nju.edu.cn

Abstract*—With the acceleration of IP traffic and urbanization, Device-to-Device (D2D) communication has appeared as a promising solution to delivering contents to mobile users in a timely manner in smart cities by offloading traffic from the overburdened cellular infrastructure. The process to distribute the new content to all the interested devices is called content synchronization. After all the interested devices have been synchronized, the existence of a large number of content copies in the network reduces resource utilization and network performance. Removing these redundant copies in D2D communication networks with multiple interested devices and no guaranteed end-to-end path between devices is challenging. In this paper, we propose two novel feedback mechanisms namely the acknowledgement-based feedback mechanism (AFM) and the non-acknowledgement-based feedback mechanism (NAFM) built into the relay-assisted synchronization algorithm under the City Section mobility model to address the issue. AFM relies on acknowledgements and NAFM relies on the estimation of synchronization latency to remove redundant copies without adding extra overhead. Simulation results show that both are effective in removing redundant copies: NAFM removes all the copies with little overhead by good estimation of the synchronization latency and AFM has a tradeoff between synchronization latency and number of copies removed.

I. Introduction

Due to the rapid increase of global IP traffic [2] and the accelerating pace of urbanization [1], Device-to-Device (D2D) communication network [4], [17] has appeared as a promising solution in a "smart" city to delivering contents such as video streaming, video gaming, video conferences, and multimedia to mobile users in a timely manner by offloading traffic from the overburdened cellular infrastructure.

D2D communication is defined as the direct communication between two mobile devices capable of short range communication [8], [12]. Thus, if a mobile device gets an updated content from the Internet via cellular, it can directly distribute this latest version to other devices interested in the content when they move into a close distance without going through the cellular network. The process to distribute the new content to all the interested devices is called *content synchronization* in these devices. However, due to the high mobility of the devices, there does not guarantee a path between the devices all the time. Hence there has been an increasing interest to operate D2D over cellular using multi-hop transmissions (henceforth referred to as *multi-hop D2D*) [6], [15]. The D2D communication we refer to in this paper is multi-hop.

Despite the research on spectrum and interference issues in D2D by the previous works [4], [6], there are very few articles discussing multi-hop D2D content synchronization strategies which are important to verify the usefulness of D2D communication to alleviate the traffic load in the cellular networks. In our previous work [16], we put forward a relay-assisted synchronization strategy under the City Section mobility model [10] that has a very low latency comparable to the epidemic strategy. It uses a constant number of relays (nodes that are not interested in the content) to help distribute the new content to all the nodes that are interested in the content. However, after all of the interested nodes have received the content, a large number of content copies still exist in the relay nodes. These redundant copies occupy the limited buffer space so that the resource utilization reduces and network performance degrades. Therefore, in the paper, we explore the methods to remove these redundant copies in the synchronization strategy in a timely-manner. Here we use the relay-assisted synchronization strategy as an example but our idea can be applied to other synchronization strategies.

In order to remove these redundant copies, an efficient feedback mechanism is required. A traditional feedback method is to let a destination send an Ack to the network after it gets the content to delete redundant copies. However, the traditional feedback mechanisms based on TCP/IP protocol are inapplicable due to the lack of an end-to-end path between nodes in D2D communication networks. In addition, there is more than one destination in content synchronization. Those feedback mechanisms addressing only one destination [3], [5], [18], [22], [23] cannot be directly applied to the multi-destination case. This is because in the one-destination case, an Ack will be sent out by the destination immediately after it receives a copy, but in the multi-destination case, an Ack sent out too early would reduce the number of copies in the network, resulting in synchronization delay for other interested nodes. Therefore, it is a challenging task to design an efficient feedback mechanism to meet the demand of fast synchronization and the improvement of resource utilization in D2D communication.

To address the above challenge, we propose two feedback mechanisms in the relay-assisted synchronization strategy for D2D. The first one is called *acknowledgement-based feedback mechanism* (AFM) and the second one is called *non-acknowledgement-based feedback mechanism* (NAFM). In AFM, we follow the idea of using acknowledgement. That is, after an interested node receives the content, an Ack will be generated and sent out. Based on how the Ack is spread, we distinguish active and passive feedback mechanisms in AFM. Different from the one-destination case, the key point here is when to send out an Ack because if an Ack is sent out too early, the reduction in the relay copies will delay the synchronization of other interested nodes. Though sending acknowledgement is a widely adopted feedback mechanism, itself also increases network traffic and resource utilization. Therefore, we put forward a novel non-acknowledgement-based method NAFM. This method relies on the estimation of the time to synchronize all the interested nodes which we refer to as the *synchronization latency*. The idea is simple: after the estimated synchronization latency time is reached, all the relay nodes will delete their copy of the content. This is our first attempt to explore the possibility of using theoretical estimation to remove redundant copies in the network without adding extra overhead.

The main contribution of this paper and its difference from other existing works are summarized as follows.

- We adopt the City Section Mobility model to mimic node movement in a city environment to discuss content synchronization.
- We design feedback mechanisms for multiple destinations, which as far as we know, have not been discussed.
- We propose an acknowledgement-based feedback mechanism AFM and a novel non-acknowledgement-based feedback mechanism NAFM that relies on the estimation of synchronization latency to reduce redundant copies.
- We conduct simulations to compare the performance of the proposed mechanisms. Simulation results show that both are effective in removing redundant copies.

The rest of the paper is organized as follows. Section II references the related works. Section III introduces the preliminaries. Sections IV and V propose the acknowledgement-based and non-acknowledgement-based synchronization algorithms, respectively. Section VI presents the simulations conducted to compare the proposed algorithms. And the paper is concluded in Section VII.

II. Related Works

There are currently two feedback mechanisms in DTN: active and passive feedbacks [13].

A. Active Feedback. After receiving the message, the destination generates a corresponding acknowledgment and actively forwards it to the encountered nodes to delete their copies in a flooding way. However, the acknowledgments themselves increase the overhead dramatically. In this category, Viet et al. [21] proposed *social routing* where a simple message called *signal message* is used to travel through the network to remove copies of the arrived messages.

B. Passive Feedback. In passive feedback, an acknowledgment will not be sent to the encountered node until the encountered node tries to relay the message to reduce the overhead caused by the acknowledgments. Ali et al. [3] designed a global selective strategy based feedback mechanism G-SACK to ensure reliable transmission. An et al. [5] proposed an adaptive feedback mechanism using both active and passive acknowledgements based on the congestion level of the network. The dynamic segment network coding scheme proposed by Zeng et al. [23] evacuates the acknowledged segment when two nodes meet to avoid redundant transmissions. Wang et al. [22] put forward a social aware feedback mechanism where acknowledgements are forwarded to the encountered nodes whose Social Link (SL) is higher than a given threshold α to remove their redundant copies.

The above feedback mechanisms are proposed for routing algorithms in DTNs and are triggered once the destination node receives the message. Different from them, we will work on the feedback methods for content synchronization involving multiple destinations in a smart city environment which is mimicked by the City Section Mobility model.

III. Preliminary

In this section, we introduce the preliminary information of our feedback mechanisms. We first describe the mobility and network models, then list the assumptions, and then describe the relay-assisted synchronization algorithm used for the feedback mechanisms.

A. Mobility Model

Most previous research on wireless mobile networks assumes that mobile nodes move freely in a given area, i.e. the Random Walk [9], [11] and Random Waypoint [14] mobility models. However, in the real world, node mobility is restricted to obstacles and rules. For example, nodes (e.g. vehicles) in a city always move along the streets. Davies et al. introduce the City Section mobility model in [10] to simulate node mobility in a street network in a city. And this model has also been widely adopted to evaluate the performance of wireless ad hoc and vehicular networks [20].

In the City Section mobility model, the street network of a city is mapped onto a grid road topology known as a *grid graph*, and the node movement is constrained to the grid. All the edges in the grid graph are considered as bi-directional, double-lane roads with speed limits. Initially mobile nodes (i.e. vehicles) are randomly deployed in the street intersections. Then each mobile node randomly selects an intersection in the grid graph as its destination and moves towards it with constant speed. We assume the movement from the current position to the destination adopts the shortest path between the two points. Upon reaching the destination, the mobile node pauses for a specified time and then randomly chooses another destination and repeats the process.

B. Network Model

We consider a D2D network of N nodes randomly deployed in a grid graph of size $L \times L$. All nodes move according to

Algorithm Relay-assisted Synchronization

Require: The source node s that has the new content and the node set interested in the content $I = \{i_1, i_2, \cdots, i_{M-1}\}$; s is the initial content holder x; the number of relay nodes allowed is K

while not all of the interested nodes receive the new content **do**
  On contact between an updated node x and a node y (when their distance is less than transmission range R):
  if $y \in I$ and is not updated **then**
    /* Reached an interested node */
    y is updated and will follow the same process as x;
  else if $y \notin I$ and $K > 0$ **then**
    /* Reached a relay candidate */
    $K = K - 1$;
    y gets the content and becomes a relay node and will follow the same process as x;
  else
    Do nothing
  end if
end while

Fig. 1. Relay-assisted synchronization algorithm

Relay-assisted Synchronization with Acknowledgement -based Feedback Mechanism (AFM)

Follow the relay-assisted synchronization in Fig. 1
if an interested node y gets the new content **then**
  it prepares an Ack with a time stamp
  if the number of Acks y has seen $>$ the percentage h multiplies the number of interested nodes **then**
    y sends its Ack to the node it encounters
  end if
end if
For **Active** feedback: the Ack is actively forwarded to the encountered nodes in a flooding way
For **Passive** feedback: the Ack will not be sent to the encountered node until the encountered node tries to relay the content
if a node receiving an Ack is a relay **then**
  it removes its copy of the content
end if

Fig. 2. Relay-assisted synchronization with AFM

the City Section mobility model. Each node has a transmission range of distance R ($R \geq 0$). Two nodes "encounter" if they are within each other's transmission range R. We assume links are bi-directional. And like [19], we assume that the transmission range R is very small compared with the city length L. Thus two nodes can communication directly only when they encounter.

C. Relay-assisted Synchronization Strategy

The basic idea of relay-assisted synchronization is to introduce a number of relay nodes to help distribute the updated content. Assume in a network with N nodes, M of them have a copy of a file (thus they are interested in the file content). When one node gets the latest version of the file, there are $M - 1$ nodes, which are called *not-updated nodes*, waiting to be synchronized. And the $N - M$ nodes without the file, which are called *candidates*, can serve as relays. Assume only $K (K \leq N - M)$ relay nodes are used. The strategy can be described as follows and the detailed algorithm is in Fig. 1.

- If an updated node encounters a not-updated interested node, they synchronize their content.
- If an updated node encounters a candidate and the relay number is less than K, it copies the updated file to the candidate and makes it a relay.
- A newly updated or a relay node will act exactly the same as an updated node.

IV. Acknowledgement-based Feedback Mechanism (AFM)

In this section, we present the relay-assisted synchronization with acknowledgement-based feedback mechanism AFM. This scheme relies on the acknowledgement sent from a synchronized node after it gets the latest content to remove the copies in the relay nodes. The algorithm follows the basic idea of relay-assisted synchronization except that after an interested node y receives the new content, it will prepare an Ack. Different from the feedback mechanisms involving only one destination, when there are multiple nodes interested in the content and if y sends out its Ack right away, then the reduction of the relay nodes in the network will delay the synchronization of the remaining interested nodes. So we only allow y to send out its Ack when the number of Acks it has seen exceeds a certain percentage h ($0 \leq h \leq 1$) of the number of interested nodes. To make sure that y does not count the same Ack twice, each synchronized node attaches a time stamp to its Ack. In the acknowledgement-based feedback mechanism, there are two ways to spread the Acks. One is Active and the other is Passive. In the Active method, an Ack is actively forwarded to the encountered nodes in a flooding way while in the Passive method, an Ack is sent to the encountered node if it is asked to relay the new content. Finally, in both Active and Passive methods, if a node receiving the Ack is a relay, then it will remove its copy of the content.

V. Non-acknowledgement-based Feedback Mechanism (NAFM)

In this section, we present the relay synchronization with non-acknowledgement-based feedback mechanism NAFM. This scheme relies on the estimation of the synchronization latency in the relay-assisted strategy. That is, after the estimated synchronization time of a content is reached, all the relay nodes will delete the copy of the content in their buffers.

The details of the mechanism is presented in Fig. 3. The idea is that when the source node s first starts to distribute the content, it attaches the starting time and the estimated latency to the content. Later, a relay node will delete its copy of the

Relay-assisted Synchronization with Non-Acknowledgement-based Feedback Mechanism (NAFM)

Require: The source node s that has the new content and the node set interested in the content $I = \{i_1, i_2, \cdots, i_{M-1}\}$; s attaches to the content the starting time of the synchronization and the estimated latency T_r; s is the initial content holder x; the number of relay nodes allowed is K

Follow the relay-assisted synchronization in Fig. 1
if node y is a relay and its current time $>$ starting time + latency T_r **then**
  y removes the content from its buffer;
end if

Fig. 3. Relay-assisted synchronization with NAFM

message at the time of the summation of the starting time and the latency. Next, we derive the estimated synchronization latency T_r in the relay-assisted strategy.

The following theorem provides the upper bound of the relay-assisted synchronization latency in D2D networks.

Theorem 1: Assume a file f is replicated in M mobile nodes out of a total of N mobile nodes in the D2D network. The maximum number of allowed relay nodes is K. When the file is updated in a node, let T_r be the expected latency of synchronizing the content in the rest $M-1$ nodes to the new version. T_e is the expected pair-wise encounter time interval. The following upper bound holds for the relay-assisted synchronization strategy:

$$\begin{aligned} T_r \leq & \tfrac{T_e}{(N-M+1)}(H_K + H_{N-M} - H_{N-M-K}) \\ & + \tfrac{T_e}{(K+M)}(H_{M-1} + H_{K+M-1} - H_K). \end{aligned} \quad (1)$$

Proof:

In the relay-assisted synchronization strategy, a maximum of K relays are used to help distribute the updated content. The analysis of its synchronization process is complicated because when an updated node encounters another node, it may update the content of the other node or make it a relay. In order to simplify the analysis, we assume that the synchronization process goes through the following two phases:

Phase 1: Spread the updated content to K relays;

Phase 2: Distribute the update to all not-updated interested nodes.

We assume phase 2 occurs after phase 1 has finished. That is, only after K relays have the updated content will the $M-1$ not-updated nodes be synchronized. Thus, the sum of the expected latencies of the two phases will give an upper bound to the latency of the strategy. Assume the expected latencies of phase 1 and 2 are T_{phase1} and T_{phase2}, respectively.

We now analyze the expected latency in phase 1. In the beginning, there was only one updated node. Assume the first K candidate nodes met by the updated nodes will become relays. After i relays are found, let $T_1(i)$ $(0 \leq i \leq K)$ be the expected synchronization latency to update the rest $K-i$ relays in phase 1. Apparently $T_1(K) = 0$ and $T_{phase1} = T_1(0)$.

Consider the event that one of the $i+1$ updated nodes encounters one of the $N-M-i$ relay candidates, the expected time is $\frac{T_e}{(i+1)(N-M-i)}$. Therefore,

$$T_1(i) = \frac{T_e}{(i+1)(N-M-i)} + T_1(i+1),$$

for $0 \leq i \leq K-1$.

That is,

$$\begin{aligned} T_1(i) - T_1(i+1) &= \frac{T_e}{(i+1)(N-M-i)} \\ &= \frac{T_e}{N-M+1}\left(\frac{1}{i+1} + \frac{1}{N-M-i}\right), \end{aligned}$$

Solving the equation, we obtain

$$\begin{aligned} T_{phase1} &= T_1(0) \\ &= \frac{T_e}{(N-M+1)}\left(\sum_{i=0}^{K-1}\frac{1}{i+1} + \sum_{i=0}^{K-1}\frac{1}{N-M-i}\right). \end{aligned}$$

If we use $H_k = \sum_{i=1}^{k}\frac{1}{i}$ to denote the *kth* Harmonic Number [7], the above result can be refined as

$$T_{phase1} = \frac{T_e}{(N-M+1)}(H_K + H_{N-M} - H_{N-M-K}).$$

In phase 2, starting from the $K+1$ updated nodes, we study the expected latency when the $M-1$ not-updated nodes are synchronized. Let $T_2(j)$ $(1 \leq j \leq M-1)$ be the expected remaining time of phase 2 after j not-updated nodes are synchronized. In this case, the expected time of any of the $K+1+j$ updated nodes encounters any of the $(M-1-j)$ not-updated nodes is $\frac{T_e}{(K+1+j)(M-1-j)}$. We have

$$T_2(j) = \frac{T_e}{(K+1+j)(M-1-j)} + T_2(j+1).$$

Apparently $T_2(M-1) = 0$ and $T_{phase2} = T_2(0)$. Solving the equation, we obtain

$$T_{phase2} = T_2(0) = \frac{T_e}{(K+M)}(H_{M-1} + H_{K+M-1} - H_K).$$

So, we obtain the following estimated upper bound:

$$\begin{aligned} T_r &\leq T_{phase1} + T_{phase2} \\ &= \frac{T_e}{(N-M+1)}(H_K + H_{N-M} - H_{N-M-K}) \\ &+ \frac{T_e}{(K+M)}(H_{M-1} + H_{K+M-1} - H_K). \end{aligned}$$

The theorem is proved. $\square$

Here we do not make any assumptions about the distribution of pair-wise encounter time interval T_e. In practice, the value of T_e can be obtained from the encounter history information of the node pairs in D2D networks.

VI. Simulations

In this section, we conduct simulations to evaluate the performance of our proposed strategies. Due to the lack of the existing tools and strategies that use the City Section mobility model and the feedback schemes that address the multi-destination case, we wrote a customized simulator in Matlab to compare the following related algorithms:

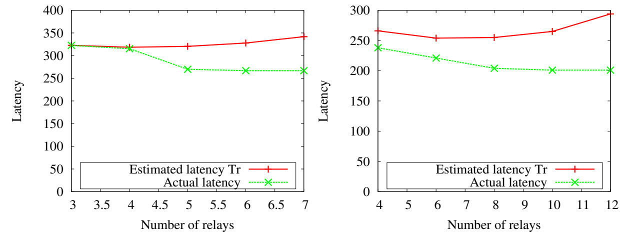
1) *The Direct Contact Synchronization Algorithm* (DIRECT): the source spreads the update through synchronization directly to every interested node it encounters.
2) *The Relay-assisted Synchronization Algorithm* (RELAY): spreads the update using a constant number of relays without considering feedback.
3) *The Relay-assisted Synchronization Algorithm with Active Feedback* (ACTIVE): the relay-assisted synchronization with active feedback.
4) *The Relay-assisted Synchronization Algorithm with Passive Feedback* (PASSIVE): the relay-assisted synchronization with passive feedback.
5) *The Relay-assisted Synchronization Algorithm with Non-acknowledgement-based Feedback* (NAFM): the relay-assisted synchronization with estimated-latency-based feedback.

We used the metrics below to compare their performance.

- *Synchronization Latency or simply Latency*: it is the total number of hops moved by the mobile nodes to synchronize all the interested nodes.
- *Forwardings*: it is the number of forwardings generated in the synchronization process.
- *Remaining copies*: it is the number of remaining copies existing in the relay nodes after all the interested nodes are synchronized.

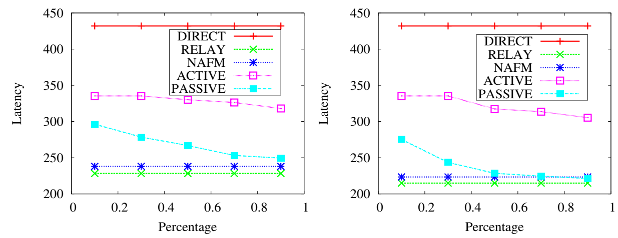
In the following experiments, we randomly generated a set of mobile nodes in a 100×100 grid and let them move according to the City Section mobility model. We set the transmission range to 5 and the time length to synchronize the content to $2,000$ so that all the interested nodes can obtain the new content during the time frame. We randomly picked one node as the source node that had the latest content, a number of other nodes interested in the content, and some relay nodes to help spread the content in the Relay-assisted algorithms. We set different values to the parameters and ran the proposed algorithms with each parameter setting $1,000$ times and averaged the results.

In the first experiment, we checked the accuracy of the estimated latency T_r using formula (1) by comparing it with the actual value. The results are in Fig. 4(a)(b). In Fig. 4(a), we set the number of mobile nodes N to 15, the number of interested nodes to 5, and varied the number of relays from 3 to 7. In Fig. 4(b), we set N to 20, the number of interested nodes to 6, and varied the number of relays from 4 to 12. The results show that our estimation provides an upperbound to the actual latency. After T_r is reached, all the copies in the relays are removed. The remaining unsynchronized interested nodes will be synchronized by the synchronized interested nodes.

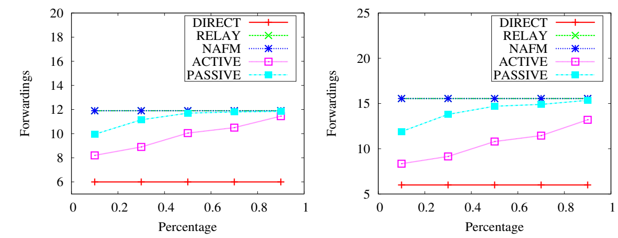


(a) $N = 15$, 5 interested nodes (b) $N = 20$, 6 interested nodes

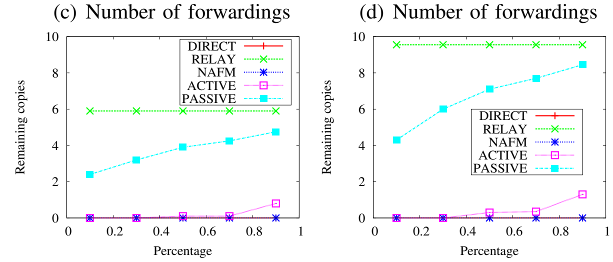
Fig. 4. Comparison of the estimated latency T_r and the actual latency



(a) Delivery latency (b) Delivery latency



(c) Number of forwardings (d) Number of forwardings



(e) Number of remaining copies (f) Number of remaining copies

Fig. 5. (a)(c)(e) Comparison of algorithms with six relay nodes; (b)(d)(f) Comparison of algorithms with ten relay nodes

In the second experiment, we compared the listed algorithms. We set N to 20, the number of interested nodes to 6 besides the source, and the number of relay nodes to 6. The simulation results are shown in Figs. 5(a)(c)(e). The x-coordinate in these three figures represents the percentage h of the interested nodes we used to remove the copies in the relay nodes in the ACTIVE and PASSIVE algorithms. Since DIRECT, RELAY, and NAFM do not involve the percentage, their performance values remain constants in these figures.

Fig. 5(a) shows the synchronization latency of the comparing algorithms. DIRECT and RELAY provide the upper and lower bounds of the latency because DIRECT does not rely on any relay and RELAY can use up to six relays without removing the copies in the relays throughout the whole synchronization process. The latency of NAFM is very close to RELAY with a value of 238 in this experiment. The estimated T_r using formula (1) is 250, which is slightly bigger than the actual latency. This means that the synchronization process can be finished within the estimated time T_r by taking full advantage of the relay nodes. The latency of ACTIVE is higher than that of PASSIVE because ACTIVE is more actively removing the copies in the relay nodes after some interested nodes are synchronized. The more the copies in the relay nodes are removed, the longer the synchronization latency. With the increase of the percentage h, the latencies of both ACTIVE and PASSIVE decrease because they will remove fewer copies in the relay nodes, resulting in more relay nodes helping synchronize the content.

Fig. 5(c) presents the number of forwardings generated in the synchronization process. DIRECT has the lowest number of forwardings six because the source synchronizes the six interested nodes directly. RELAY and NAFM have the same highest number of forwardings (their lines are overlapped in the figure) because NAFM did not remove any copies during T_r in this experiment and both RELAY and NAFM can fully use these relay nodes. ACTIVE has fewer number of forwardings than PASSIVE because it actively removes relay nodes so the forwardings from the relays are reduced. With the increase of the percentage h, the number of forwardings in both ACTIVE and PASSIVE increase because they have more forwardings from the relays by removing fewer copies.

Fig. 5(e) shows the remaining copies in the relays after the synchronization process has finished. Both DIRECT and NAFM have zero copies in the relays afterwards because DIRECT never uses any relays and NAFM removes all the copies in the relay after T_r expires. RELAY has the highest number of copies (six) in the relays because it never removes them. ACTIVE has much fewer remaining copies than PASSIVE because it is more active in removing them. With the increase of the percentage h, the number of remaining copies in both ACTIVE and PASSIVE increases because they remove fewer copies in the relay nodes.

In our next experiment, we increased the number of relay nodes from 6 to 10 and the results are shown in Fig. 5(b)(d)(f). These results exactly match those of the six-relay-node case.

In summary, the results show that both AFM and NAFM are effective in reducing redundant copies. NAFM removes all the copies with little overhead by good estimation of the synchronization latency and AFM has a tradeoff between synchronization latency and number of copies removed.

VII. Conclusion

In this paper, we have proposed two novel feedback mechanisms namely the acknowledgement-based feedback mechanism (AFM) and the non-acknowledgement-based feedback mechanism (NAFM) built into the relay-assisted synchronization algorithm using City Section mobility model to address the issue of redundant copies existing in the network after interested nodes have been synchronized. Simulation results have shown that both mechanisms are effective in removing redundant copies. As we have seen, the accuracy of the estimated synchronization latency is critical to the success of NAFM. Based on our first attempt here, in the future, we will continue exploring the estimation methods and make our model better reflect the D2D communication environment.

Acknowledgment

This research was supported in part by NSF under CNS1305302, ACI1440637, and China NSF grant 61373128.

References

[1] The World Fact Book. https://www.cia.gov/library/publications/the-world-factbook/fields/2212.html.

[2] The Zettabyte Era: Trends and Analysis. http://www.cisco.com/c/en/us/solutions.

[3] A. Ali, T. Chahed, E. Altman, M. Panda, and L. Sassatelli. A new proposal for reliable unicast and multicast transport in delay tolerant networks. In *Proc. of PIMRC*, pages 1129–1134, 2011.

[4] R. Alkurd, R. Shubair, and I. Abualhaol. Survey on device-to-device communications: Challenges and design issues. In *IEEE International New Circuits and Systems Conference*, 2014.

[5] Y. An, J. Huang, H. Song, and J. Wang. A Congestion Level based end-to-end acknowledgement mechanism for Delay Tolerant Networks. In *Proc. of GLOBECOM*, pages 1574–1579, 2012.

[6] A. Asadi, Q. Wang, and V. Mancuso. A survey on device-to-device communication in cellular networks. *IEEE Communication Surveys & Tutorials*, 16(4), 2014.

[7] S. Baase and A. Gelder. *Computer Algorithms: Introduction to Design and Analysis*. Addison Wesley, 3 edition edition, 1999.

[8] A. Balasubramanian, R. Mahajan, A. Venkataramani, B. N. Levine, and J. Zahorjan. Interactive wifi connectivity for moving vehicles. In *ACM SIGCOMM*, 2008.

[9] S. Basagni, I. Chlamtac, V. R. Syrotiuk, and B. A. Woodward. A distance routing effect algorithm for mobility (dream). In *MobiCom'98*, pages 76–84, New York, NY, USA, 1998. ACM.

[10] V. Davies. Evaluating mobility models within an ad hoc networks. Master's thesis, Colorado School of Mines, 2000.

[11] A. Einstein. *Investigations on the Theory of the Brownian Motion*. Dover Publications, 1956.

[12] J. Eriksson, H. Balakrishnan, and S. Madden. Cabernet: vehicular content delivery using wifi. In *Proc. ACM MobiCom*, pages 199–210, 2008.

[13] K. A. Harras and K C. Almeroth. Transport layer issues in delay tolerant mobile networks. In *Proc. of the 5th international IFIP-TC6 conference on Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications Systems*, pages 463–475, 2006.

[14] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, pages 153–181. Klumer Academic Publishers, 1996.

[15] B. Kaufman, J. Lilleberg, and B. Aazhang. Spectrum sharing scheme between cellular users and ad-hoc device-to-device users. *IEEE Transactions on Wireless Communications*, 12(3):1038–1049, 2013.

[16] W. Li, X. Chen, and S. Lu. Content Synchronization using Device-to-Device Communication in SMART Cities. Technical report, Nanjing University, 2015.

[17] J. Liu, N. Kato, J. Ma, and N. Kadowaki. Device-to-device communication in lte-advanced networks: A survey. *IEEE Communication Surveys & Tutorials*, 16(4), 2014.

[18] D O. Seo and D H. Lee. *The end-to-end reliability algorithms based on the location information and implicit ACK in delay tolerant mobile networks*. Springer Berlin Heidelberg, 2011.

[19] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Efficient routing in intermittently connected mobile networks: The multiple-copy case. *IEEE/ACM Transactions on Networking*, 16(1):77–90, Feb. 2008.

[20] C. Tracy, J. Belong, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communication & Mobile Computing(WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2(5):483–502, May 2002.

[21] Q. Viet, P. Phuoc, T. Son, and L. Duc. Social Routing: A Novel Routing Protocol for Delay Tolerant Network based on Dynamic Connectivity. In *Proc. of IEEE International Conference on Electro/Information Technology (EIT)*, pages 415 – 420, 2015.

[22] K. Wang, G. Huang, L. Shu, C. Zhu, and L. He. A Social Awareness based Feedback Mechanism for Delivery Reliability in Delay Tolerant Networks. In *Proc. of ICC*, pages 7007–7011, 2015.

[23] D. Zeng, S. Guo, and J. Hu. Reliable bulk-data dissemination in delay tolerant networks. *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, pages 2180–2189, 2014.