Data Synchronization in Ad Hoc Mobile Networks

Tianxiang Li, Spyridon Mastorakis, Xin Xu, Haitao Zhang, Lixia Zhang
UCLA
tianxiang,mastorakis,xinxu129,haitao,lixia@cs.ucla.edu

ABSTRACT

Intermittent connectivity and dynamic network topology create unique challenges for distributed applications in Mobile Ad Hoc Networks (MANETs), where individual entities may produce data at any time while moving around continuously. In this poster, we present DDSN, a distributed dataset synchronization protocol in Named Data Networking (NDN) that enables applications in MANETs to keep all members in an application group synchronized on the latest state of a shared dataset. Taking intermittent connectivity as the norm, whenever nodes in the same application group are reachable to each other, DDSN provides an efficient mechanism to unify the shared state and dataset between the encountered nodes.

CCS CONCEPTS

• **Networks → Transport protocols**;

KEYWORDS

NDN, MANET, Sync

ACM Reference format:
Tianxiang Li, Spyridon Mastorakis, Xin Xu, Haitao Zhang, Lixia Zhang. 2018. Data Synchronization in Ad Hoc Mobile Networks . In *Proceedings of 5th ACM Conference on Information-Centric Networking, Boston, MA, USA, September 21–23, 2018 (ICN '18),* 2 pages.
DOI: 10.1145/3267955.3269024

1 INTRODUCTION

Mobile Ad Hoc Networks (MANETs) are infrastructure-less wireless networks formed by mobile nodes as they move into each other's communication range. The movement of nodes in MANET results in intermittent connectivity and network partition, which introduce challenges for applications in demand of efficient and reliable data synchronization among multiple parties. For example, in the disaster relief scenario such as the aftermath of an earthquake, where network infrastructure has been damaged, rescuers working on the field need to synchronize information regarding survivors, supplies, and damage status in order to collaborate in rescue efforts.

In this poster, we present a solution for distributed dataset synchronization in MANET based on the Named Data Networking (NDN) architecture. NDN proposes a data-centric communication model in contrast to IP's host oriented model. In NDN, each piece of data is uniquely named and signed at the time of generation, which enables name-based fetching of data from the network regardless of its location. Our solution, dubbed Distributed Data Synchronization over NDN (DSSN) protocol enables a group of communicating nodes in MANET to efficiently exchange knowledge about the latest data generated by all the nodes in the group, and fetch its missing data from connected neighbors in the area.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
ICN '18, Boston, MA, USA
© 2018 Copyright held by the owner/author(s). 978-1-4503-5959-7/18/09…$15.00
DOI: 10.1145/3267955.3269024

2 DDSN DESIGN

The goal of DDSN is to achieve distributed dataset synchronization in MANET where intermittent connectivity is the norm. We assume each node in the network can move around continuously, acting as both a data producer and data consumer. We refer to all the nodes which want to synchronize data with each other as *members* in the same *sync group.* All members in the sync group are registered under the same multicast group prefix. Each member maintains a local state of knowledge about the latest data produced by all the members in the sync group, known as *dataset state.*

The dataset state in DDSN is encoded using a version vector called *state vector* (Section 2.1). The exchange of state vectors allow members to directly deduce the dataset state mismatches, and infer the names of each missing data piece through pre-defined naming conventions, without any assumptions on the consumer/producer state as in the previous NDN Sync protocols [5]. Members then send Interests to fetch each missing data piece. We refer to the exchange of state vectors and the fetching of missing data together as the *sync process.*

Due to the mobility of nodes in MANET, members need to discover neighbors within its communication range in order to initiate the sync process (Section 2.3). To do so, members periodically send *Beacon Interests* containing a cryptographic hash of its dataset state known as *state digest* [7] [1]. The Beacon Interest itself is used to both detect neighbors and detect dataset state mismatches. When a member receives a beacon Interest containing a state digest representing a dataset state different from its own, it initiates the sync process (2.2). When members do not hear any transmissions for a certain period, it is a sign of isolation, and it falls back to periodic beacon transmissions.

2.1 State Vector

The State Vector represents a member's knowledge of the dataset state. Figure 1 shows an example representation of the dataset state, and the encoded state vector. DDSN adopts the sequential naming convention adopted by ChronoSync [7]. Since the data sequence number increases monotonically, the latest data generated by each producer can be represented by its producer name prefix plus its latest sequence number. The state vector encodes the producer name prefix of each member in the sync group and its latest sequence number in a version vector [6]. The state vector also avoids the need to maintain a separate membership list as done in VectorSync [4].

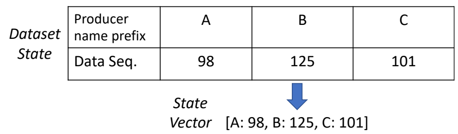

Figure 1: State Vector

2.2 Data Sync Process Overview

As shown in Figure 2, the sync process is initiated by a member sending a Sync Interest containing its state vector to all other members within its vicinity. This can be triggered either passively by the reception of beacon Interest containing mismatched state digest, or proactively by the generation of new data while detecting there are neighbors in its vicinity. When a member receives a Sync Interest, it would first send back a Sync Reply packet containing its state vector to express its dataset state and satisfy the pending Sync Interests in the PIT of the Sync Interest sender. Due to the one Interest one Data transmission of NDN, when multiple members receive the Sync Interest, the Sync reply adopts a delay timer based on the degree of state difference. In cases where multiple members meet, the Sync Reply would trigger the Sync Interest transmission by another member until all members have exchanged its state vector or until certain transmission limit is reached. Based on the state vectors exchanged, members then prioritize on the data to fetch to best satisfy the other members within the vicinity. To ensure efficiency of the data fetching process under short connectivity, DDSN adopts encapsulation of multiple Interest/Data packets, with intact signatures for each data packet.

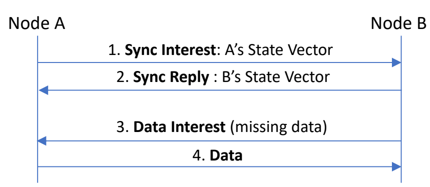

Figure 2: Dataset Sync Process Overview

2.3 Neighbor Discovery

As members are moving around, it is important to detect other newly encountered members to learn whether they may have a different dataset state; besides new data they may have generated themselves, they may also carry data generation information from other partitions that they passed by. To discover new neighbors and their dataset state, each node sends a Beacon Interest periodically that carries a digest of the sender's newest dataset state.

When a member receives a Beacon Interest: (i) the receiver cancels its upcoming beacon transmission for the current beacon interval, and (ii) compares state digest with its own to detect state inconsistency with the sender. If the state is different, the receiver initiates the sync process. Once the members in the connected neighborhood reaches a stable state, they fall back to Beacon Interest transmission, to periodically check the consistency of states among neighbors. Only one member within the vicinity sends out a Beacon Interest in each beacon interval.

In Figure 3, we present an example of a member entering a new partition group. A Beacon Interest sent by node D is received by node C that is already a member of the partition group. Node C compares the digest value in the received beacon with its own digest, and determines that it has a different dataset state with node D, thus deciding to sync with D. Subsequently, C sends a Sync Interest containing its updated State Vector to further propagate the information in the partition group (to node B and then A).

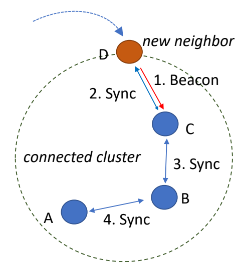

Figure 3: Updating Reachable Neighborhood

3 CONCLUSION AND FUTURE WORK

Based on lessons learned from previous NDN Sync protocols, we introduce the design of DDSN which not only provides robust distributed data synchronization in MANET but also works for networks with stable connectivity. We have implemented a DDSN prototype in C++, and used ndnSIM simulator [3] to perform extensive evaluations of its performance. For the next step, we plan to implement DDSN on mobile devices and gather results from real-world users to test out its performance. We also plan to investigate the commonalities/differences between DDSN and peer-to-peer data sharing solutions, such as nTorrent [2].

4 ACKNOWLEDGEMENT

This work is partially supported by the National Science Foundation under awards CNS-1629922 and CNS-1719403.

REFERENCES

[1] Pedro de-las Heras-Quirós, Eva M Castro, Wentao Shang, Yingdi Yu, Spyridon Mastorakis, Alexander Afanasyev, and Lixia Zhang. 2017. *The design of RoundSync protocol.* Technical Report. Technical Report NDN-0048, NDN.
[2] Spyridon Mastorakis, Alexander Afanasyev, Yingdi Yu, and Lixia Zhang. 2017. nTorrent: Peer-to-Peer File Sharing in Named Data Networking. In *26th International Conference on Computer Communications and Networks (ICCCN).*
[3] Spyridon Mastorakis, Alexander Afanasyev, and Lixia Zhang. 2017. On the evolution of ndnSIM: An open-source simulator for NDN experimentation. *ACM SIGCOMM Computer Communication Review* 47, 3 (2017), 19–33.
[4] Wentao Shang, Alexander Afanasyev, and Lixia Zhang. 2017. VectorSync: distributed dataset synchronization over named data networking. In *Proceedings of the 4th ACM Conference on Information-Centric Networking*. ACM, 192–193.
[5] Wentao Shang, Yingdi Yu, Lijing Wang, Alexander Afanasyev, and Lixia Zhang. 2017. *A Survey of Distributed Dataset Synchronization in Named-Data Networking.* Technical Report NDN-0053. NDN Project.
[6] Xin Xu, Haitao Zhang, Tianxiang Li, and Lixia Zhang. 2018. Achieving Resilient Data Availability in Wireless Sensor Networks. In *2018 IEEE International Conference on Communications Workshops (ICC Workshops).* IEEE.
[7] Zhenkai Zhu and Alexander Afanasyev. 2013. Let's chronosync: Decentralized dataset state synchronization in named data networking. In *Network Protocols (ICNP), 2013 21st IEEE International Conference on.* IEEE, 1–10.