Large-scale peer-to-peer network for mobile platforms: Challenges and Experiences

Nirmit Desai
IBM T. J. Watson Research Center
Yorktown Heights, NY, USA
nirmit.desai@us.ibm.com

Wendy Chong
IBM T. J. Watson Research Center
Yorktown Heights, NY, USA
wendych@us.ibm.com

Heather Achilles
IBM T. J. Watson Research Center
Yorktown Heights, NY, USA
hachilles@us.ibm.com

Shahrokh Daijavad
IBM T. J. Watson Research Center
Yorktown Heights, NY, USA
shahrokh@us.ibm.com

Thomas La Porta
Dept. of Computer Science and Engineering
Penn State University
University Park, PA, USA
tlp@cse.psu.edu

***Abstract*—Peer-to-peer (p2p) networks and Mobile ad hoc networks (MANET) have been widely studied. However, a real-world deployment for the masses has remained elusive. Ever-increasing density of mobile devices, especially in urban areas, has given rise to new applications of p2p communication. However, the modern smartphone platforms have limited support for such communications. Further, the issues of battery life, range, and security remain unaddressed. A key question then is, what kinds of applications can the modern mobile platforms support and what challenges remain? This paper describes our experiences in deploying a real-world peer-to-peer delay-tolerant network to millions of users for relaying severe weather information, the challenges faced, and the approach taken.**

***Index Terms*—peer-to-peer systems, mobile ad hoc network, delay-tolerant network**

I. Introduction

MANETs have been greatly studied given their decentralized nature and potential for new applications [2], [5]. Most of the prior work has focused on valuable analytical and simulation-based study of MANET behavior [3], [4], [6]. Given the outstanding practical challenges of physical nodes, real-world implementations have been limited and have not reached mass scale [1]. However, with the growth of smartphones, large-scale real-world implementations may become feasible. Mobile devices with programmable platforms such as android and iOS have steadily grown over the last decade, surpassing the 2 billion mark [1]. This paper describes a real-world implementation of a p2p delay-tolerant network, called Mesh Network Alerts (MNA), for relaying severe weather information to millions of mobile device users as part of the Weather Channel app[2]. Specifically, we describe the key challenges in such an implementation, approaches taken to address them, experimental results, and a characterization of the outstanding challenges.

II. Practical Challenges

We focus on three main challenges and describe a system implementation for addressing them. One, even when a user is not interacting with the weather app, or worse yet, when the device is not being used at all, the devices must actively discover peer devices to send and receive potentially life-saving weather information. Although modern mobile operating systems such as Android and iOS offer APIs to discover and advertise information to peer devices over WiFi and Bluetooth interfaces, peer-to-peer connections do not work when the same APIs are accessed while the app is in the background. Prior works, widely document these challenges and take the approach of having special access on the devices, e.g., jail-breaking or rooting []. We overcame these challenges via innovative techniques, without resorting to hacks that may violate user security or Appstore guidelines. With our technique, exchange of information happens without making network connections. As we describe later, this is achieved by splitting the messages into small enough chunks so that they can be stuffed into service advertisements themselves and broadcast over multiple advertisements in a quick succession.

Two, since devices may be offline when new weather information arrives, MNA on each device must remain active at all times to be able to discover new information as soon as it arrives. However, the MNA activity must keep the device battery consumption to a minimum. This is a challenge because advertising and discovery are power-hungry operations over the radio channels. Our approach here is two-pronged. Firstly, via extensive experiments on a large number of devices, we fine-tune the algorithm governing the intervals at which discovery and advertisements occur. In a nutshell, receiving new information during a period causes MNA to be more aggressive in discovery and advertisement. Similarly, lack of new information for a period makes the device less aggressive. Secondly, we allow "wake up" messages to be broadcast in the network ahead of an anticipated severe weather event. When devices receive such messages, they schedule themselves to

DAIS-ITA

[1] https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/

[2] https://weather.com/apps/ibm/meshnetworkalerts

remain aggressive during the specified window of time. Outside of this window, the device can afford to have long sleep cycles and conserve power. With these techniques, our testing shows less than 1% battery consumption per hour on most device models.

Three, given the heterogeneity of devices and operating system distributions in the market, it is quite a challenge to test whether MNA functions as expected on a given device. Further, running test scenarios on a p2p network at large-scale is non-trivial given that a large number of devices need to take specific coordinated action followed by coordinated observations to determine whether a test passes or fails. This is not a challenge in traditional mobile application development as the software is confined within a single connected device. We developed test automation tools and processes such that multiple devices can be controlled from a single test station and follow prescribed steps to generate, send, and receives messages to play out a test scenario. Finally, the framework allows automated analysis of the observations to determine the test result. This capability was instrumental in uncovering bugs at a fast pace to meet the aggressive timeline and avoid the cost of acquisition. Further, the automation framework is general and can be expressly applied to test other apps in this fashion.

Lastly, as devices with mesh advertise on a continuous basis, it may be possible for a malicious attacker to listen for such advertisements and reverse-engineer the message formats and protocols used. Then, the attackers may generate fake messages and advertise them, e.g., a fake tornado alert. Such false messages cannot be distinguished from the real ones, and MNA will end up propagating them to as many devices as possible, "poisoning" the network. In general, veracity of such information cannot be independently verified in open decentralized distributed systems and previous work on peer-to-peer networks do not address this challenge. In our approach, since the origin of weather information is the Weather Channel service, digital signatures can be attached to all weather alerts broadcast from the service. The mobile application is distributed with the corresponding public key so that digital signatures from the service can be verified. If a message fails such a verification, it is discarded and not forwarded any further.

III. System Architecture

IV. Experimental Results

Acknowledgment

References

[1] W. Kiess and M. Mauve. A survey on real-world implementations of mobile ad-hoc networks. *Ad Hoc Networks*, 5(3):324 – 339, 2007.

[2] J. Loo, J. L. Mauri, and J. H. Ortiz. *Mobile Ad Hoc Networks: Current Status and Future Trends*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 2011.

[3] S. Marti, T. J. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, MobiCom '00, pages 255–265, New York, NY, USA, 2000. ACM.

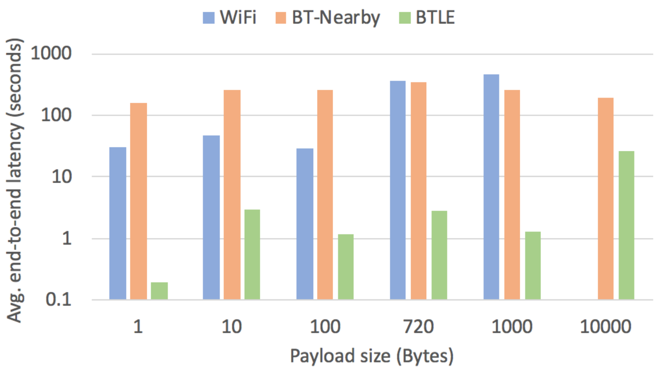

Fig. 1. End-to-end latency

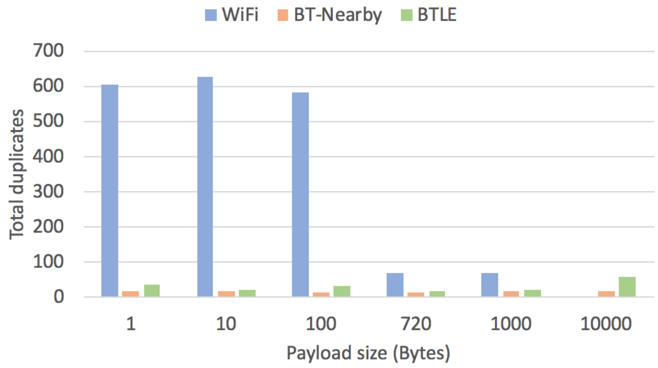

Fig. 2. Duplicity due to flooding

[4] M. Mauve, J. Widmer, and H. Hartenstein. A survey on position-based routing in mobile ad hoc networks. *IEEE Network*, 15(6):30–39, Nov 2001.

[5] C. E. Perkins. In *Ad Hoc Networking*, chapter Ad Hoc Networking: An Introduction, pages 1–28. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.

[6] X. M. Zhang, Y. Zhang, F. Yan, and A. V. Vasilakos. Interference-based topology control algorithm for delay-constrained mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 14(4):742–754, April 2015.

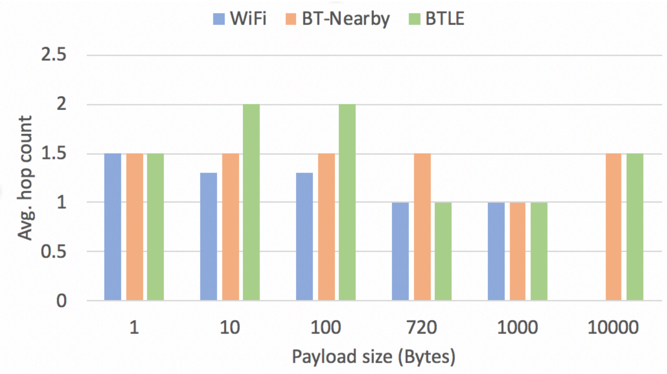

Fig. 3. Average hop count