International Journal of Research (IJR)
e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 10, October 2015
Available at http://internationaljournalofresearch.org

Multi Incentive Based Mechanism for Data Sharing In Delay Tolerant Mobile Networks

[1]**Shaik Neeha**
Dr.K.V.Subba Reddy College of Engg for Women, Kurnool, Andhra Pradesh
[2]**D.Wasiha Tahseen**
Asst.Professor, Dr.K.V.Subba Reddy College of Engg for Women, Kurnool, A.P

Abstract:

Work centers on data dissemination in delay-tolerant mobile networks, where data fall into a range of interest types and each node may have one or multiple interests. The goal is to deliver data messages from sources to nodes with corresponding interests. The key challenge is to effectively track the value of a message under such a unique network setting with intermittent connectivity and multiple interest types. Moreover, a message is usually desired by multiple mobile users. Therefore, it can be potentially "sold" multiple times to different receivers. On the other hand, while more than one copies can be created during the transmissions of a message, a particular receiver "pays" for the first received copy only. These characteristics together make the development of Multi Incentive mechanism a unique, interesting, and challenging problem. In this paper, we present effective schemes to estimate the expected credit reward, and formulate nodal communication as a two-person cooperative game, whose solution is found by using our Theorem. Extensive simulations are carried out based on real-world traces to evaluate the proposed scheme in terms of data delivery rate, delay and overhead. To our best knowledge, this is the first work that incorporates incentive stimulation into data dissemination in delay-tolerant mobile networks with selfish nodes and multiple interest types.

Keywords: Tolerant Mobile Networks; Data Sharing; Multi Incentive Mechanism

***

1. INTRODUCTION

Modern distributed applications are increasingly designed as collections of components interacting through an event-based communication paradigm. Some components observe events and *publish* notifications, which are in turn delivered to those components who *subscribed* to receive them. In the *content-based* incarnation of this publish subscribe model, the filtering of relevant events is specified by the subscriber using predicates on the event content (e.g., using regular expressions and logic operators), therefore providing additional expressiveness and flexibility. Content-based publish-subscribe has proven useful in a number of distributed scenarios, due to its high degree of decoupling, a synchronicity, and versatility. Moreover, its inherently reactive style of communication makes it particularly suited for highly dynamic scenarios like those characterizing mobile ad hoc networks (MANETs), peer-to-peer networks, and sensor networks.

The potential for the content-based publish-subscribe *model*, however, can be fully unleashed in these scenarios only if the underlying *system* is designed in a way that is compatible with their requirements. Highly dynamic settings like the aforementioned ones pose unprecedented challenges, mostly determined by the fluidity of the system's network topology. Mainstream systems implementing a distributed infrastructure for publish-subscribe are typically geared to large-scale settings, striving for scalability by routing events on a tree-shaped overlay network,but with few or no mechanisms in place to deal with topological reconfiguration. Our research group has been particularly active in studying how to design content-based publish-subscribe systems able to efficiently tolerate frequent topological reconfigurations. Nevertheless, the approach we followed thus far also relied on the existence of a tree-shaped overlay network. Indeed, this provided a good starting point that enabled us to build up on well-established results in the field. However, depending on the degree of dynamicity and other parameters characterizing the scenario, maintaining a tree overlay may bring additional overhead and complexity. On the other hand, approaches in closely related fields, e.g., MANET multicast and subject-based publish-subscribe, cope with dynamicity by routing messages probabilistically, instead of deterministically based on the collection of subscription information. In doing so, they effectively trade delivery guarantees for enhanced scalability, fault tolerance, and resilience to topological changes.

In this paper, we take a different perspective and establish a clear point of departure from our previous work—and, to the best of our knowledge, from that of other researchers' as well. First, our solution relies on an undirected connected graph topology. Given the dynamicity requirements of our target scenarios, a graph structure is not only considerably easier to maintain than a tree, but also provides opportunities for more fault-tolerant solutions by intrinsically providing multiple routes between any two dispatchers. Second, our solution is neither entirely deterministic or probabilistic. Instead, it strikes a balance between the two, since it combines the efficiency of deterministic routing with the resilience to reconfiguration and inherent simplicity of probabilistic approaches.

The core of our approach is very simple. Subscriptions are propagated only in the immediate vicinity of a subscriber, in contrast to most existing systems. Event routing leverages of this subscription information, whenever available, by deterministically routing an event along the link a matching subscription was received from. If no subscription information exists at a given dispatcher, events are forwarded along a randomly chosen subset of the available links. Simulations show not only that our approach is successful per se in providing high delivery rates with low over head, but also confirm that it performs better than a fully deterministic (or probabilistic) alone. Moreover, nice byproducts of our strategy are a significant reduction of the routing tables size, and easier development and understanding of the actual implementation.

The rest of the paper is organized as follows. Section 2 provides the reader with a concise overview of content-based publish-subscribe. Section 3 presents the details of our approach, by illustrating the strategy for routing subscriptions and events. Section 4 provides an evaluation of our approach through simulation in several scenarios exhibiting different degrees of dynamicity, including mobile ones. Section 5 places our work in the context of related ones. Finally, Section 6 concludes the paper and hints at opportunities for further research on the topic.

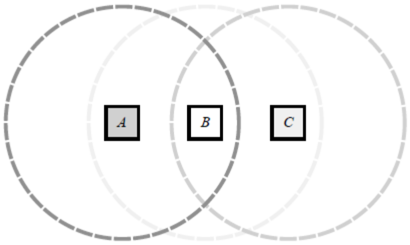

Figure 1 A simple ad hoc network of three wireless mobile hosts

2. OPPORTUNISTIC EXCHANGE

2.1 Procedure Description

Denote by r the wireless transmission range. We say that two vehicles *encounter* each other when their distance is smaller than r. When two vehicles A and B encounter each other, A and B first exchange their resources. Upon receiving new resources, vehicle A computes the relevance for each received resource and re-evaluates the relevance of its own resources. If all the resources do not fit in *A*.s memory space, the least relevant ones are purged. If two moving objects travel within the transmission range for a period of time, after the initial exchange only newly arrived resources are exchanged. We assume that if *A* encounters two or more vehicles simultaneously, the exchanges occur sequentially. In other words, we assume that there is a mechanism to resolve interference and conflicts.

2.2. Spatial and Temporal Boundaries of Resource Distribution

In this subsection, we theoretically analyze the opportunistic exchange procedure, and show that with the above procedure, a resource is always propagated within a bounded area, and there exists an age threshold beyond which the resource disappears from the system. In the following analysis we assume that the wireless transmission range r is negligible and the time consumed by each resource exchange is negligible. Denote by M the memory allocation, and by v the maximum speed a vehicle can travel with. We say that a resource is *rejected* by O at time t if O receives but does not save the resource at t. A resource is *new* for a time interval [t1,t2] if the resource is created during this time interval.

3. PERFORMANCE EVALUATION

3.1 Simulation Setup

To evaluate the performance of our protocol, we perform extensive simulations using a simulator we have developed in C. Simulation experiments are performed with 1000 nodes randomly placed across a campus with a size of one square kilometer. A snapshot of the simulated environment is shown in Figure 2. A data file of size 1MB is used for dissemination. Each supplying peer has a transmission rate of _ 128K bps. _ _ (of the peer population are randomly assigned as initial supplying peers, while □__ (of the peer population are randomly assigned as requesting peers. All the peers are assumed to be cooperative. It is also assumed that to re-construct a data file of size 1MB, a requesting peer will need 1.055MB data due to the decoding inefficiency factor _. The transmission range of each peer is 20 meters. One run of the simulation lasts for 500 seconds. Initially each peer is positioned in a building or on a street. Initial coordinates within building (corresponding to a room) or on street are randomly determined. If a peer is placed in a building, it stays static for a certain wait period. If a peer is placed on a street, it is assigned a velocity according to the mobility model explained earlier.

Our simulation progresses in a second-by-second fashion.

During each second, the following steps are performed by every supplying peer or a non-supplying but collaborative peer willing to re-broadcast the encoded packets:

- Determine the set of non-supplying peers (neighbors) within its transmission range. These are the peers that will receive the packets it broadcasts.
- For a supplying peer, randomly select a packet - previously unsent, from the set of encoded packets to broadcast to all its neighbors. It maintains an array of the encoded data packets and cycles through them.
- For a non-supplying peer which is relaying packets, randomly select a packet from the packets received previously but not yet relayed, and broadcast it to all its neighbors. Note that such a peer has fewer number of encoded packets than a supplying peer.

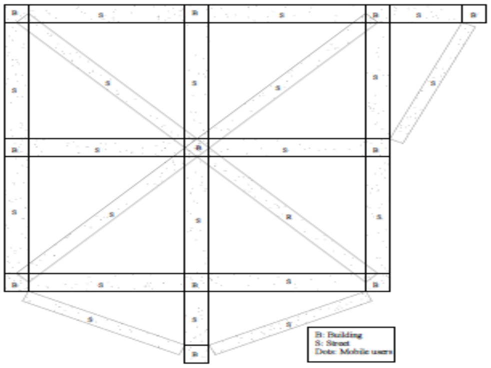

Figure 2. A snapshot of the simulated environment

3.2 Simulation Results

Figure 4 compares the progress of data dissemination
with and without Tornado encoding. For the simulation without Tornado encoding, the 1MB data file is divided into 400 packets, each with a size of 2.5KB. All these 400 packets have to be received by each requesting peer in order to restore the file. For the simulation with Tornado encoding, a stretch factor of 2 is used. Therefore, 800 packets of size 2.5KB are generated, out of which 422 (_ _ _ ____) packets are needed by each requesting peer to correctly re-construct the file. We observe that the first requesting peer becomes ‘infected’ after 25 seconds in the Tornado encoding simulation, while it takes more than 100 seconds to infect the first requesting peer in the simulation without Tornado encoding. Almost all the requesting peers become ‘infected’ after 150 seconds in the Tornado encoding simulation, while it takes around 450 seconds in the simulation without Tornado encoding to infect all the requesting peers. This experiment demonstrates the efficiency of Tornadocoding-based dissemination of popular content in a mobile ad-hoc environment.

Figure 5 shows the progress of Tornado-coding-based data dissemination, under various packet sizes. As in the previous experiment, the 1MB data file is stretched to 2MB of encoded data under a stretch factor of 2. However, this may be done by generating 800 packets of size 2.5KB, 200 packets of size 10KB, 40 packets of size 50KB, or 20 packets of size 100KB. The transmission rate is 128Kbps. For packets of size 2.5KB or 10KB, they can be transmitted in an overlapping period as short as 1 second. However, packets of size 50KB can be fully transmitted only when the overlapping period is at least 4 seconds; and packets of size 100KB require an overlapping period of at least 7 seconds. Recall that a packet is deemed useless if it cannot be received in full by a requesting peer. This explains why the curves that correspond to packet sizes 2.5KB and 10KB grow faster than that for the 50KB packet size, which in turn grows faster than the curve for the 100KB packet size. We also notice that the curve for 2.5KB packet size is slightly steeper than the curve for 10KB packet size. This can be explained as follows: suppose the average overlapping period between a supplying peer and a requesting peer is 1 second, then six 2.5KB packets, or 15KB of data can be transmitted, while only one 10KB packet can be fully transmitted during the same period. Similarly, if the overlapping period is 3 seconds, then nineteen 2.5KB packets, or 47.5KB of data can be transmitted fully, versus only four 10KB packets or 40KB of data fully transmitted. This experiment shows that bandwidth wastage is reduced with decrease in packet size. However, excessively small packet size will lead to a large number of packets and thus increase the Tornado decoding overhead.

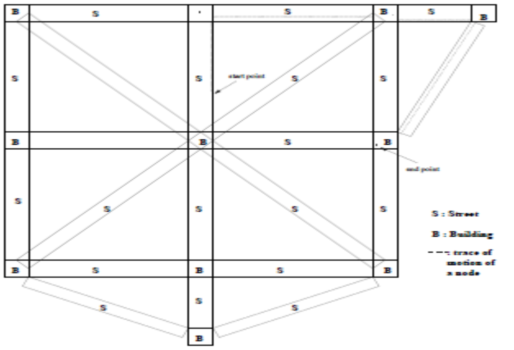

Figure 3. A trace of the motion of a particular peer during the simulation.

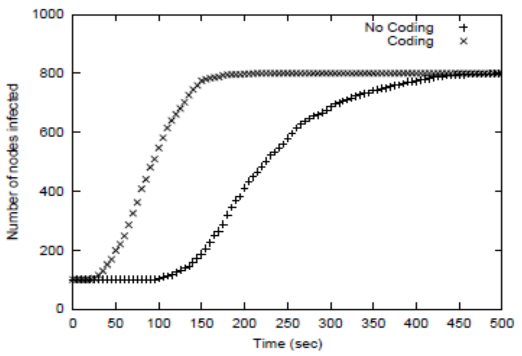

Figure 4. Progress of data dissemination: with and without Tornado coding

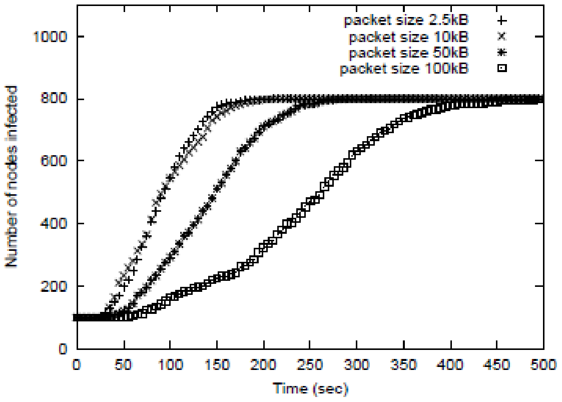

Figure 5. Progress of data dissemination: under different packet sizes

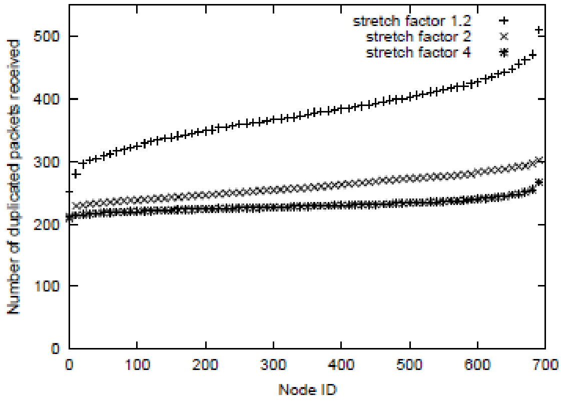

Figure 6. Number of duplicate packets received by each requesting peer: under different stretch factors

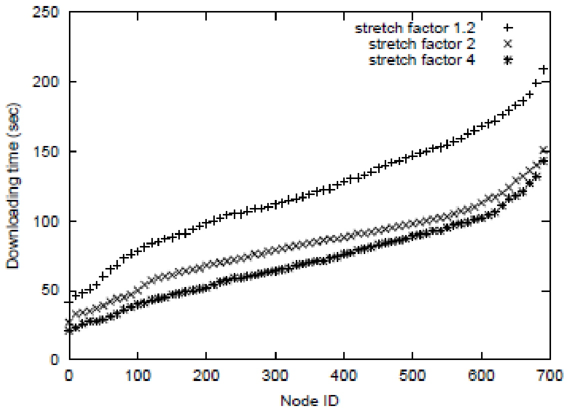

Figure 7. File downloading time of each requesting peer: under different stretch factors

4. RELATED WORK

In this section, we discuss the related work previously proposed to improve data accessibility in mobile (not necessarily ad hoc) networks. Kubach *et al* propose a prefetching approach to accessing data in wide-area wireless networks [6]. Their approach predicts the information needed in the future, and hoards this information on the mobile device from Info-stations before the user begins roaming. The limitation of this approach is that it assumes the information is location-dependent, and that the information will be requested only when the user is close to the corresponding location. Future requests are predicted by learning from the history of mobility/request patterns of the same or other users, assuming a recurring pattern in user preference and motion. The amount of information hoarded is limited by the memory size of mobile devices. Meanwhile, their approach assumes a wide deployment of Info-stations, which may not be true in practice. Various mobility models have been proposed for mobile ad hoc networks [9, 8, 5]. Hong *et al* have proposed a Reference Point Group Mobility Model in which groups of mobile nodes are formed based on their geographical proximity [5]. Wang *et al* have proposed a Reference Velocity Group Mobility Model in which groups of mobile nodes are formed based on the relative velocities of the nodes [9]. In both [9] and [5], each peer must know the identities of all group members. In terms of information exchange, this may prove quite expensive. The frequency at which group membership information is exchanged, as well as the paths along which group membership information is propagated require careful consideration. Papadopouli *et al* have adopted a Random Walk Model for mobility modeling in their simulation experiments [8]. The Random Walk Model may not be suitable for highly regulated civilian environment, because it assumes that a mobile node has equal probability of moving in *any* direction. In [4], Hara proposes a placement and partition scheme for replicated data access in wireless ad hoc networks.

It focuses on the provision of high data availability in a partitionable ad hoc network, so that each mobile node can access a copy of the data it needs with high probability. However, it does not address the issue of dynamic data dissemination, especially in a peer-to-peer fashion. Byers *et al* first propose Tornado coding [1, 2, 3, 7]. Their scheme involves the encoding of a large file (of k packets) to n encoded packets (where n = stretch factor * k). The original file can then be recovered by decoding _ _ _ arbitrary but distinct encoded packets. They apply Tornado coding to parallel file downloading from multiple *static* hosts, while we propose to apply Tornado coding to data dissemination in mobile ad hoc environments. We argue that the selection of Tornado coding parameters has to reflect the mobility model of the targeted ad hoc environment.

5. CONCLUSION

In this paper, we propose the application of Tornado coding to data dissemination in mobile ad-hoc networks, with the objective of enabling efficient and reliable peer-topeer data sharing among mobile users. Our solution consists of (1) a Street-and-Building mobility model suitable for modeling mobile users in a regulated civilian environment and (2) a peer-to-peer data dissemination protocol to dissemination Tornado encoded file segments (packets). We discuss the impact of Tornado coding parameters on the performance of our peer-to-peer data dissemination protocol. Our simulation results show satisfactory performance of the protocol: it reduces the file downloading time of a requesting peer by as much as 75%. The simulation results also demonstrate the importance of mobility-aware Tornado coding parameter selection.

6. REFERENCES

[1] L. Yin and G. Cao, “Supporting cooperative caching in ad hoc networks,” *IEEE Trans. Mobile Comput.*, vol. 5, no. 1, pp. 77–89, Jan. 2006.

[2] J. Zhao, S. Member, Y. Zhang, S. Member, G. Cao, and S. Member, “Data pouring and buffering on the road: a new data dissemination paradigm for vehicular ad hoc networks,” *IEEE Trans. Veh. Technol.*, 2007.

[3] H. Zhou and S. Singh, “Content based multicast (CBM) in ad hoc networks,” in *Proc. 2000 Workshop on Mobile and Ad Hoc Networking and Computing*, pp. 51–60.

[4] Vincent Lenders, Martin May, and Gunnar Karlsson. Wireless Ad Hoc Podcasting. In *Proceedingsof IEEE SECON*, San Diego, CA, June 2007.

[5] Martin May, Clemens Wacha, Vincent Lenders, and Gunnar Karlsson. Wireless Opportunistic Podcasting: Implementation and Design Tradeoffs. In *Proceedings of the ACM MobicomCHANTS Workshop*, Montreal, Canada, September 2007.

[6] M. Nottingham and R. Sayre (Eds.). The Atom Syndication

[7] K. Fall, “A delay-tolerant network architecture for challenged internets,” in *Proc. 2003 ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, pp. 27–34.

[8] S. Burleigh, A. Hooke, L. Torgerson, K. Fall, V. Cerf, B. Durst, K. Scott, and H. Weiss, “Delay-tolerant networking: an approach to interplanetary internet,” *IEEE Commun. Mag.*, vol. 41, no. 6, pp. 128–136, June 2003.

[9] T. Spyropoulos, K. Psounis, and C. Raghavendra, “Spray and wait: an efficient routing scheme for intermittently connected mobile networks,” in *Proc. 2005 ACM SIGCOMM Workshop on Delay-Tolerant Networking*, pp. 252–259.

[10] Y. C. Hu, M. D. Saumitra and H. Pucha, “Exploiting the Synergy between Peer-to-Peer and Mobile Ad Hoc Networks”, *Proc. of HotOS-IX*, Hawaii, May 2003.

[11] K. G. Siddhartha, M. Singh, D. Xu, and B. Li, “Efficient Peer-to-Peer Data Dissemination in Mobile Ad-Hoc Networks”, *Proc. of International Workshop on Ad Hoc Networking*, Vancouver, BC, August 2002.

[12] M. Papadopouli, and H. A. Schulzrinne, “Performance Analysis of 7DS a Peer-to-Peer Data Dissemination and Prefetching Tool for Mobile Users”, *Advances in wired and wireless communications, IEEE Sarnoff Symposium Digest*, Ewing, NJ, March 2001.

[13] G. Kortuem, *et al.*, “When Peer-to-Peer Comes Face-to-Face: Collaborative Peer-to-Peer Computing in Mobile Adhoc Networks”, *Inter. Conference on Peer-to-Peer Computing*, Linköpings, Sweden, August 2001.