

Reputation system for user-generated podcasting under community based mobility model

Liang Hu
Technical University of Denmark
2800 LYNGBY, Denmark
liang.hu@fotonik.dtu.dk

Lars Dittmann
Technical University of Denmark
2800 LYNGBY, Denmark
ladit@fotonik.dtu.dk

ABSTRACT

A user-generated podcasting service over mobile opportunistic networks can facilitate the user generated content dissemination while humans are on the move. However, in such a distributed and dynamic network environment, the design of efficient content forwarding and cache management schemes are challenging due to the lack of global podcast channel popularity information at each individual node. We design a distributed reputation system at each node for estimating the global channel popularity information which is significant for forwarding and cache management decision. Our simulation result shows that, compare to History-based rank scheme, our reputation system can significantly improve system performance under Community-based Random Way-Point (C-RWP) mobility model and localized channel popularity distribution. The performance evaluation under three C-RWP scenarios shows that the reputation system brings more performance gain when channel popularity distribution becomes more localized and node mobility become more localized.

Categories and Subject Descriptors

C.2.4 [Computer System Organization]: Computer Communication Networks-Distributed Systems; I.6 [Computing Methodologies]: Simulation and Modeling

General Terms

Algorithms, Performance, Design

Keywords: Reputation system, ad-hoc podcasting, User Generated Service, Bayesian Framework

Permission to make digital or hard copies of all or part 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. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WICON'08, November 17-19, 2008, Maui, Hawaii, USA.

Copyright 2008 ICST 978-963-9799-36-3.

1. INTRODUCTION

In recent years, opportunistic network has become an attractive research area for networking small mobile devices carried by human being, vehicles and animals. Besides unicast routing, dissemination based routing is proposed by PodNet project [1] to provide seamless content distribution beyond infrastructure network. This dissemination based routing particularly support applications in which the set of user interested in receiving a given data is not known in advance. In this paper, we focus on designing reputation-based content forwarding and cache replacement schemes for User Generated Wireless Podcasting (UGWP) service over the system architecture of PodNet. We mainly target at obsolete podcasting service where only the most recent content is of interests and old content is always obsolete by the latest one e.g. short news report distribution or software updates of mobile devices. In UGWP, obtaining popularity information of podcast channels is significant for the content forwarding and cache replacement decisions. Unlike existing Internet-based user generate service such as YouTube [2] where the content popularity information is made centralized, in ad-hoc podcasting, the channel popularity information is fully distributed throughout the network and dynamic due to nodes' mobility. Thus it is much more difficult for each node to obtain and predict popularity information of global channels. With inaccurate channel popularity information, node may forward the content that future encounter nodes are not interested in. Ultimately, this would lead to low hit ratio of content retrieve, low utilization of both the node contact opportunities and cache storage.

In this paper, we design a distributed reputation system based on Bayesian framework through which each node can estimate the global channels popularities. The popularity of channel is represented by the reputation rating. The reputation system consist of three parts: Firstly, the reputation rating of channels at each node is built and updated by the number of requests to each channel from encounter nodes. This is called the first hand information of channel popularity by each node's direct observations. Secondly, reputation rating is also updated by integrating its encounter nodes' direct observations which is called the second hand information of channel popularities. By doing so, node can learn and adjust popularity information of channels from observations made by others even before having to learn by own experience. By nodes gossiping the channel reputations, the accurate channel popularity information can

propagate much faster throughout the network, especially when the popularity distribution is non-uniform and localized. Moreover, to protect against rumor spread from liars, the second hand information is only accepted if a deviation test is passed. Thirdly, to adapt the channel popularity shifts, both the first hand information and the reputation ratings of each channel decays after each contact. The previous observations are gradually forgotten while more weight is put on recently observations.

To the best of our knowledge, our work is the first attempt to employ Bayesian Framework based reputation system for estimating the content popularity in the context of user-generated opportunistic content dissemination. Previous, the Bayesian framework based reputation system has been employed in coping with misbehaviors in mobile ad hoc networks [3]. The paper is organized as follows: in section 2, the concept of Bayesian Framework based Reputation is introduced. In section 3, the protocol specification and data structure of reputation system is described. We evaluate the performance of reputation system by discrete event simulation in section 4. Section 5 concludes the paper.

2. REPUTATION SYSTEM

To implement Bayesian framework based reputation system, both first hand information and reputation ratings are needed. First hand information is the direct observations of channel popularity and can be passed to other nodes as second hand information. Reputation rating is the channel popularity information taking accounts both first hand information and second hand information by node's encounter nodes' direct observations. In this section, we introduce how both first hand information and reputation rating is built and updated.

2.1 Standard Bayesian Framework

Node i model the popularity of channel j as an actor in the base system as follows. Node i thinks that there is a parameter θ such that the channel i is interested by any node with probability θ . The outcome is drawn independently from observation to observation (node i thinks there is a different θ for different channel j while different node i may have different believe in different parameter θ). The parameters θ are unknown, and node i model this uncertainty by assuming θ itself is drawn according to a distribution (the "prior") that is updated as new observations become available. We use Beta (A, B) as the prior distribution since it is suitable for Bernoulli distribution and the conjugate is also a Beta distribution. The standard Bayesian procedure is as follows. Initially, the prior is Beta (1,1), the uniform distribution [0,1]; this represents absence of information about which θ will be drawn. Then after ($f+s$) observations during contacts with encounter nodes, say with s times the channel i is requested by encounter nodes while f times it is not requested by encounter nodes. The prior is updated:

$$A := A + s, \quad B := B + f.$$

If θ , the true unknown value is constant, then after a large number m of contacts:

$$A \approx n\theta, \quad B \approx n(1-\theta)$$

and Beta (A, B) becomes closes to a Dirac at θ , as expected. We denote $E(\text{Beta}(A, B))$ as the expectation of Beta (A, B). Thus we can estimate θ as follows:

$$\theta \approx E(\text{Beta}(A, B)) = \frac{A}{A+B}$$

2.2 First hand information by modified Bayesian approach

The first hand information for the popularity of channel j at node i is defined as:

$$F_{i,j} = (A_j^i, B_j^i)$$

This represents the parameters of the Beta distribution assumed by node i in its Bayesian view of the popularity of channel j as an actor in the base system. Initially, it is set to (1, 1). The standard Bayesian method gives the same weight to each observation regardless of its time of occurrence. However, the popularity of a podcast channel may change when nodes move between different communities with different channel popularity distribution. For this reason, we add a reputation fading mechanism to give less weight to the past observations, because the latest observations would be more important for estimating current and future popularity of the channel. Assume node i makes one individual observation of channel j during a contact with encounter node. Let $s=1$ if channel j is requested by the encounter node, and $s=0$ otherwise. The update is as follows:

$$A_j^i := u \bullet A_j^i + s, \quad B_j^i := u \bullet B_j^i + (1-s)$$

The weight u is a discount factor for the past experiences, which serves as the fading mechanism.

2.3 Reputation Rating and Model Merge

The reputation rating of channel j at node i is defined as:

$$R_{i,j} = (\alpha_j^i, \beta_j^i)$$

Initially, it is set to (1, 1). It is built and updated on two types of events: (1) when first-hand information is updated by own observations; (2) the second hand information from encounter nodes are accepted and copied. There are two variant of using second hand information from encounter nodes: direct observations (first hand information) from encounter nodes and reputation rating from encounter nodes. For event type (1), the update of reputation rating is the same for the first-hand information updating. Let $s \in \{0, 1\}$ is the observations:

$$\alpha_j^i := u \bullet \alpha_j^i + s, \quad \beta_j^i := u \bullet \beta_j^i + (1-s)$$

For the case (2), if we assume passing direct observations, the linear pool model is used to merge own reputation rating with direct observations passed from encounter nodes on the condition if the deviation test is passed. Deviation test is used to protect system against false rating from encounter nodes. The idea behind

it is that humans only believe the opinions from others only if, to them, it seems likely i.e. it dose not differ too much from their own opinions. Moreover, even if they accepted opinions from others, they only attach less weight to other's opinions than their own opinions. Let

the first hand information of channel j at encounter node x:

$$F_{x,j} = (A_j^x, B_j^x)$$

The deviation test is as follows:

$$\text{If } |E(\text{Beta}(\alpha_j^i, \beta_j^i) - E(\text{Beta}(A_j^x, B_j^x)))| < \text{THS},$$

(THS is a positive constant (deviation threshold)), then the deviation test is passed and we believe the report from node x is trustworthy. Then, α_j^i, β_j^i are updated by first hand observations of node x using the linear pool model merging:

$$\alpha_j^i = \alpha_j^i + w \bullet A_j^x; \beta_j^i = \beta_j^i + w \bullet B_j^x, 0 < w < 1.$$

3. DATA STRUCTURE AND PROTOCOL SPECIFICATION

The cache at each node consists of a private cache (for storing node's private or own interested channels) and a public cache (for storing other nodes' interested channels). Each node maintains a table of channel reputation ratings which is used for content forwarding and public cache replacement decisions. As an example, the reputation rating table of node A is as showed in table 1:

In brief, the protocol specification of reputation system based podcasting is as follows:

1. Idle node periodically broadcast association requests to its neighbors. If it discovers several neighboring nodes, it randomly selects one node to associate and establish pair-wise connection.
2. Node updates its reputation ratings of all channels by merging the second hand information from peer if the deviation test is passed [Event 1].
3. Node firstly pulls content of private interested channels [Event 2].
4. Node updates both first hand information and reputation rating of channels by peer's requests of privately interested channels [Event 3].
5. Node pull content of public interested channels based on estimated channel popularities and popularity-based forwarding and public cache replacement schemes [Event 4]. Various forwarding and public cache replacement schemes are described later on.

For detailed description of protocol specification, see the message sequence chart at fig 1 (suppose node A and node B establish pair-wise association.).

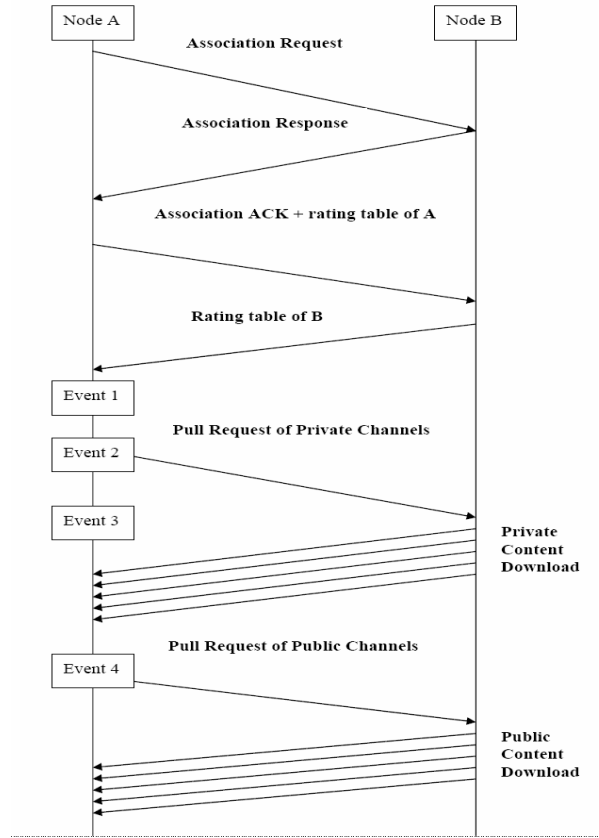


Figure 1: Message Sequence Chart

Table 1: Reputation Rating Table

Reputation rating table at node A

Channel Feeds	First Hand Information	Reputation Rating	Latest Entry ID (entry name, time of publish)	Subscribed or Not (S/N)
1	A_1^A, B_1^A	α_1^A, β_1^A	-Weather forecast of Copenhagen -13th July at 10:00 am	S
3	A_3^A, B_3^A	α_3^A, β_3^A	-BBC news -10th July at 10:00 am	S
5	S
7	S
9	S
.....	N
M	A_M^A, B_M^A	α_M^A, β_M^A		N

Channel Feeds: Podcast channel identifications.
 A_M^A, B_M^A : the first hand information of channel M at node A.
 α_M^A, β_M^A : the reputation rating of channel M.
Latest Entry ID: the latest updates correspond to each podcast channel.
"S": the channel is subscribed by node A.
"N": the channel is not subscribed.

4. PERFORMANCE EVALUATION

In this section, by discrete event simulation, we evaluate the performance of reputation system under “Community-based Random Way-Point” (C-RWP) mobility model and localized channel popularity distribution.

C-RWP captures the “clustering” effect of realistic human mobility: The mobility of nodes tends to be localized in certain geographical area where they frequently meet other nodes with similar social roles e.g. workmate, classmate; On the other hand, nodes only occasionally meet nodes with dissimilar social roles in other geographical areas. In C-RWP, nodes are divided into different communities. One community is a group of nodes with the similar mobility patterns. Nodes of one community move within the same square in a random way-point (RWP) model. Nodes of the same square have equal chance of meeting each other regularly while nodes of different squares can seldom meet each other or only occasionally meet near the borders of two squares.

Secondly, we assume channel popularity distribution. Based on the measurement results of YouTube, a recent paper [5] shows that: video clips of local interests only have a high local popularity; there is no correlation observed between global and local popularity. Along the line of their observations, we assume: firstly, one community of nodes have one group of interesting channels which is a subset of total global available channels. Among one community, the popularity of its group of channels follows Zipf-like distribution. Secondly, different communities have different groups of interested channels. One example could be one community is interested in the channels of English language while other is interested in channels of German language.

Thirdly, the location of the channel publishing nodes and its subscribing nodes could be as follows: (1) the publishing node and its subscribing are in the same community; (2) they are in two different communities which are partially or totally physically separated; (3) publishing node and some of its subscribing node are in the same community while other subscribing nodes are in other community. We focus on the scenario (2): due to physical separation of communities, nodes of one community may have difficulty of learning popularities of channels published from other communities.

4.1 Simulation Model

The simulator is based on a simple communication model: two nodes can communicate with a nominal bit-rate if their geometric distance is smaller than a threshold value. We do not model any MAC layer issues such as collision or interference, since we assume networks are sparsely connected where collisions and interference between different associations are rare. Nodes only associated pair-wise, even if more than two are within reach of one another. The reason is that the contact duration may be short and it is better to get high throughput by only sharing the transmission capacity between two parties than to get high connectivity. We assume the forwarding scheme is “Most” and

public cache replacement scheme is also “Most”. This combination gives the best performance under the ideal knowledge of channel popularity at each node [6]. The channel popularity at each node is represented by local reputation ratings. By “Most” forwarding scheme, node forward the content from the most popular channels to least popular channels until two nodes get disconnected because of their mobility or when both nodes do not have content to exchange. By “Most” public cache replacement scheme, when public cache is full, the content of less popular channel is always replaced with content of more popular one. Other simulation parameters are summarized in table 1.

Table 2

Parameters of Reputation System	
THS	0.4
u	0.99
w	0.2
Other Parameters	
Cache size	2 GB
Public Cache size	60 MB
Chunk size	2 MB
Simulated time	12 hours

4.2 Performance Metrics

To quantify the user satisfaction of user generated podcasting, the recall and delay are employed as the performance metrics of reputation system. Recall is defined as the fraction of node’s own interested chunks that are successfully received. It is borrowed from the area of Information Retrieve (IR). Delay is defined as the latency between the time when chunk is published and the time when it is received. We believe, for obsolete pocast service, both recall and delay are important for the end user satisfaction. Recall of node i by time t is defined as:

$$R^i(t) = \frac{X_R^i(t)}{X_P^i(t)}, i = 0, 1, 2, \dots, N-1$$

N : the total number of nodes; i : the node ID. $X_R^i(t)$: the total number of private interested chunks that have been received by node i by time t . $X_P^i(t)$: total number of private interested chunks that have been published from all node i ’s interested channels by time t . Average recall is defined as the average recall over the total number of nodes N . In this work, we are only interested in the average recall at the end of the simulation. Delay is defined as $\Delta t = T_{publish} - T_{receive}$. $T_{publish}$ is the chunk publish time while $T_{receive}$ is the time when it is received. M is defined as the total number of chunks received by all nodes at the end of simulation. The average delay is defined as:

$$\frac{\sum_i \Delta T_i}{M}, i = 1, 2, 3, \dots, M$$

4.3 Simulation Results

We compare the performance of Reputation System with History-based Rank [1] under three scenarios: 1. two

separated communities of nodes and two groups of localized popular channels. 2. four separated communities and two groups of localized popular channels. 3. four separated communities and four groups of localized popular channels. The history-based rank method [1] is a method which estimate channel popularity only by first hand information (in the form of number of encounter requests per channel). It works as follows: node keeps track of the channels that were requested by past encounter nodes and maintains a history-based ranking. Only the requests for channels that encounter nodes subscribed are counted.

Scenario 1: two separated communities of nodes, two groups of localized popular channels

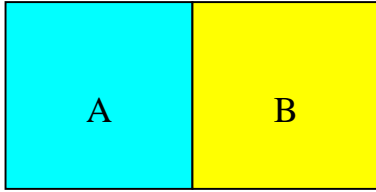


Figure 2: Scenario 1

As indicated in figure 2, 100 nodes are grouped into two communities: A (blue) and B (yellow). The nodes are human beings who carry WiFi-enabled mobile device. Each community is interested in one group of popular channels among total 100 channels. Nodes of ID 0-49 belong to community A while nodes of ID 50-99 belong to community B. Both nodes of community A and B move within a square of the same side length 500 meters in Random Way-Point (RWP) model. The moving speed is constant 1 m/s with pause time 1 s. Each node publishes one channel, with the channel ID identical to the node ID, e.g. node 0 publish channel 0, node 1 publish channel 1. Community A publish channels from 0-49 while community B publish channel from 50-99. The content publish interval per channel is 600 s which is identical for all channels. Community A is only interested in the channels published from community B (channel ID 50-99) while community B is only interested in the channel published from community A (channel ID 0-49). Each node is interested two channels. Among community B, the popularity distribution of channels 0-49 follows Zipf-like distribution with $a=1.5$, where the channel 0 is the highest popular channel, channel 1 is the second popular and so on. Define the popularity of channel 0-49 in community B:

$$P_i \sim \frac{1}{(i+1)^a}, i = 0, 1, 2, \dots, 49$$

Likewise, among community A, the popularity distribution of channels 50-99 follows the same Zipf-like distribution with $a=1.5$. Assume the channel 50 is the highest popular channel, channel 51 is the second popular and so on: Define the popularity of channel 50-99 in community A:

$$Q_j \sim \frac{1}{(j-49)^a}, j = 50, 51, 52, \dots, 99.$$

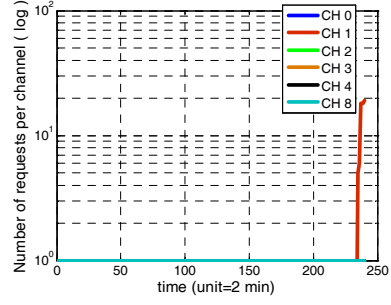


Figure 2: Number of requests per channel at node 60

From the figures 2 and 3, it is obvious that the history-based rank poorly estimates the popularity of channel 0,1,2,3,4,8. With history-based rank, node 60 cannot get any popularity information of channel 0,1,2,3,4,8 until 460 minutes. The reason is that node 60 cannot have enough first-hand information about channel popularity. In contrast, reputation system can always perfectly estimate the popularity of channel 0,1,2,3,4,8 since the very beginning of the simulation as showed in figure 3.

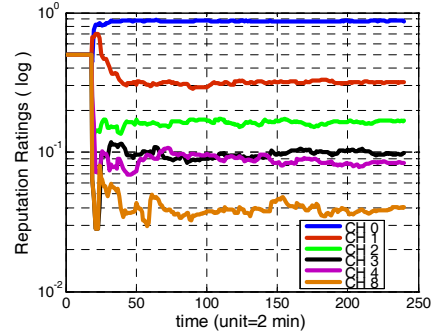


Figure 3: reputation ratings per channel at node 60

Table 3

	History-based Rank	Reputation System
Average Recall	0.015	0.250
Average Delay	1510 s	1112 s

Without the enough popularity information, nodes will not be able to forward the channels of content which are interested by its future encounter nodes. Thus the average recall is much lower when using History-based Rank method than Reputation System, as showed in the table 2. History-based Rank only achieves average recall 0.015 while Reputation System achieves 0.250. The performance gain of reputation system is more than 20 times. In terms of average delay, Reputation system also performs better than History-based Rank, where reputation system achieves almost 400s of average delay less than history-based rank As showed in table 3,

Scenario 2: four communities, two groups of localized popular channels

As indicated in figure 4, nodes are moving within four identical square areas (communities) (A1, A2, B1, and B2). Popular channels are grouped into two tastes (the red and the blue). Community A1 and A2 (red color) are only interested in channels of 50-99 published by community B1 and B2, while community B2 and B1 (blue color) are only interested in channel published by A1 and A2. Node 0-24 are moving within A1 square; node 25-49 are moving within A2 square; node 50-74 are moving within B1 square; node 75-99 are moving within B2 square. Similar to the previous scenario, each node publishes one channel. The channel ID is identical to the node ID. The channel popularity distribution among community B1 and B2 together follows Zipf-like distribution with $a=1.5$ (published from community A1 and A2). Assume the channel 0 is the highest popular channel; channel 1 is the second popular and so on. Define the number of nodes interested in channel i :

$$P_i \sim \frac{1}{(i+1)^a}, i = 0, 1, 2, \dots, 49$$

Likewise, the channel popularity distribution among community A1 and A2 together follows Zipf-like distribution with $a=1.5$ (published from community B1 and B2). Assume the channel 50 is the highest popular channel, channel 51 is the second popular and so on: define the number of nodes interested in channel j :

$$Q_j \sim \frac{1}{(j-49)^a}, j = 50, 51, 52, \dots, 99.$$

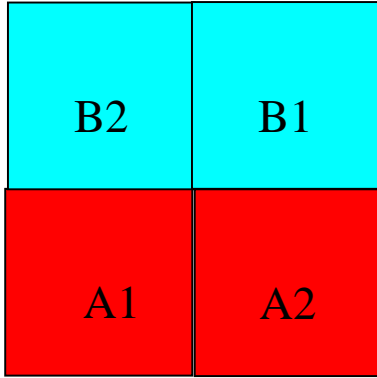


Figure 4: Scenario 2

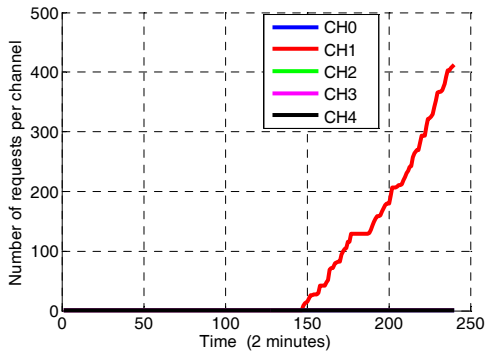


Figure 5: History-based Rank: Number of requests per channel at node 60

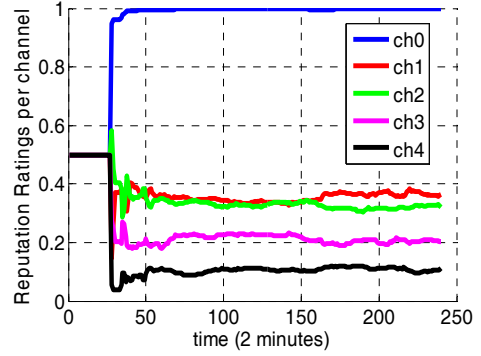


Figure 6: Reputation system: reputation ratings per channel at node 60

From figure 5 and 6, in terms of channel popularity estimation, it is obvious that reputation system far outperforms history-based rank in both in estimation accuracy and estimation speed. In figure 4, before 300 minutes, node 60 has no observations of the channel popularity information of channel 0, 1, 2, 3, 4. Even after 300 minutes, except channel 0, 1, node 60 still does not have popularity observations of other channels. In contrast, using reputation system, only after 54 minutes, node 60 can already accurately estimate the popularity ranking of channel 0, 1, 2, 3, 4, as in figure 6.

We compare the performance of reputation system with history-based rank under the impact of publish interval. Simulation Parameters are as follows: Zipf- $a=1.5$, Public Cache Size=30 chunks, Channel of Interests=2, Length of Square=350 m, Number of Channels=100.

From figure 7 and 8, we observe that, in terms of average recall, the publish interval does not have impact on the performance of history-based rank scheme. When increasing publish interval from 300s to 900s, the average recall increases slightly from 2.0 % to 6.3%. In contrast, in the case of using reputations system, the average recall increases dramatically from 0.132 to 0.390 when the publish interval increases from 300s to 900s. In terms of average delay, the reputation system method always has a much lower average delay than history-based rank.

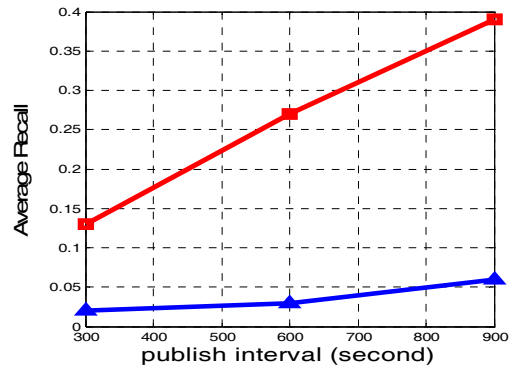


Figure 7: Average Recall

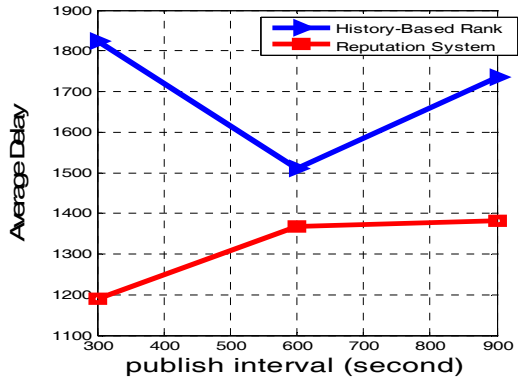


Figure 8: Average Delay

Scenario 3: four communities, four groups of popular channels

As shown in figure 9, nodes are grouped into four communities: A, B, C and D. Nodes of ID 0-24 are moving within square A area according to random way-point mobility model. Nodes of ID 25-49 are moving within square B area according to random way-point mobility model. Nodes of ID 50-74 are moving within square C area with random way-point mobility model. Nodes of ID 75-99 are moving within square C area with random way-point mobility model. The four squares A, B, C, D are all identical. Each node publishes one channel with the same ID as the node ID. The community A is only interested in the channels published by community C; the community B is only interested in the channels published by community D; the community C is only interested in the channels published by community A; the community D is only interested in channels published by community B. The channel popularity distribution of channels published from each community follows Zipf-like distribution with $\alpha=1.5$.

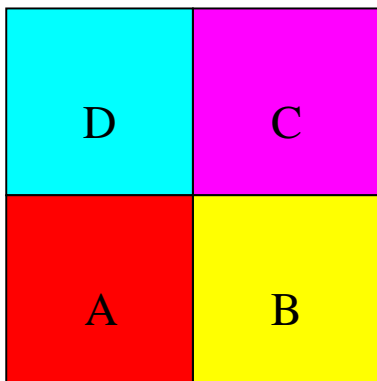


Figure 9: four communities with four groups of popular channels

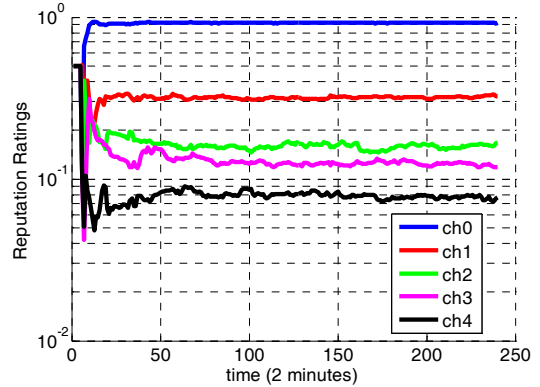


Figure 10: Reputation system: reputation ratings per channel at node 60

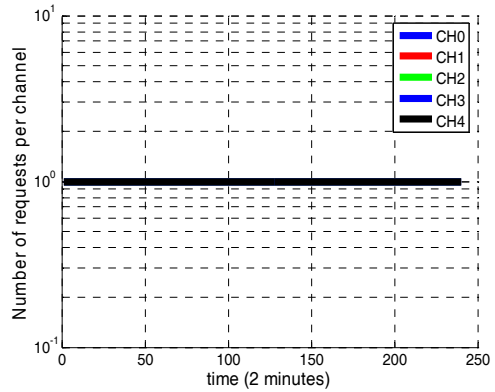


Figure 11: History-based Rank: Number of requests per channel at node 60

From figure 10 and 11, we observe that, by using history-based rank, node 60 cannot get any observations for estimating the channel popularities. In contrast, with reputation system, the channel popularity ranking information is very high and accurate. For the popularities of channel 0,1,2,3,4 has been perfectly estimated since the start of the simulation.

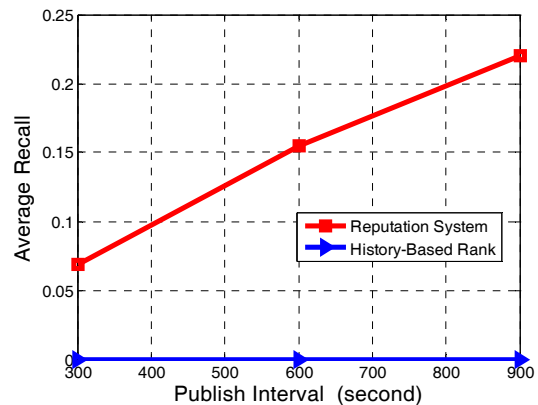


Figure 12: Average Recall

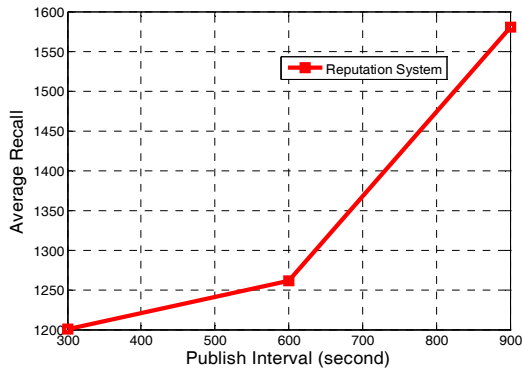


Figure 13: Average Delay

As showed in figure 12 and 13, with four communities, under different publish intervals, history based rank almost always achieves 0 average recall. With reputation system, the average recall increases from 0.069 to 0.220 when the publish interval changes from 300s to 900s. In terms of average delay, for reputation system, the delay increases from 1200 s to 1580 s as the publish interval increases from 300 to 900 s. For history-based rank, there is no average delay since 0 chunks are successfully delivered.

4.4 Summary of the simulation results

From scenario 1, 2 to scenario 3, the popularity distribution becomes more localized (i.e. from two groups of localized channels to four groups), provided the number of channels and number of nodes are the same for all scenarios. In this case, the reputation system gives more performance gain over history-based rank as the channel popularity becomes more localized. On the contrary, from scenario 1 to scenario 2, reputation system does not bring more performance gain over history-based rank as the node mobility becomes more localized (i.e. from two community to four community), given the channel popularity distributions are the same.

5. CONCLUSION AND FUTURE WORK

We design a distributed reputation system for estimating podcast channel popularity information in user generated wireless podcasting service. With reputation system, by nodes sharing their direct observations of channel popularities, the accurate channel popularity information can propagate much faster throughout the network, especially when the node mobility is community based and channel popularity distribution is localized. Our simulation results shows reputation system overwhelmingly outperforms history-based rank scheme in terms of average recall and average delay under a two-community C-RWP model and localized channel popularity distribution. Besides, the more localized the channel popularity is, the more performance gain can reputation system achieve over history based rank.

As the next step, we envision studying the performance of reputation system under a more realistic mobility model such as [4] which captures node moving both within the communities and between communities. Also, we intend to study the impact of liars on the performance of reputation system in user generated ad hoc podcasting.

6. REFERENCES

- [1] Vincent Lenders, Martin May, and Gunnar Karlsson. Wireless Ad Hoc Podcasting. In Proceedings of IEEE SECON, San Diego, CA, June 2007.
- [2] <http://www.youtube.com/>
- [3] Sonja Buchegger, Jean-Yves Le Boudec. A Robust Reputation System for P2P and Mobile Ad-hoc Networks. In Proceedings of Second Workshop on the Economics of Peer-to-Peer Systems, Harvard University, June 2004.
- [4] Mirco Musolesi, Cecillia Mascolo, Design Mobility Models based on Social Network Theory. Journal of Mobile Computing and Communication Review, Volume 11, Number 3.
- [5] Michael Zink, Kyoungwon Suh, Yu Gu and Jim Kurose, Watch Global, Cache Local: YouTube Network Traces at a Campus Network - Measurements and Implications," In Proceedings of 2008 IEEE MMCN, San Jose ,CA, January 2008
- [6] Liang Hu, Jean-Yves Le Boudec, Reputation System for wireless ad-hoc podcasting. Technical Report, Technical University of Denmark, July 2008