Computing and Informatics, Vol. 33, 2014, 965–991

OPTIMAL CACHING POLICY OF STOCHASTIC UPDATING INFORMATION IN DELAY TOLERANT NETWORKS

Yahui Wu, Mingxing Liu, Su Deng, Hongbin Huang

Science and Technology on Information Systems Engineering Laboratory
National University of Defense Technology
Changsha, China
e-mail: wuyahui@nudt.edu.cn, sudeng@suhoo.com, hbhuang@nudt.edu.cn

Yiqi Deng

Department of Computer Science
University College London
London, WC1E 6BT, UK

Abstract. To increase the speed of information retrieval, one message may have multiple replicas in Delay Tolerant Networks (DTN). In this paper, we adopt a discrete time model and focus on the caching policy of stochastic updating information. In particular, the source creates new version in every time slot with certain probability. New version is usually more useful than the older one. We use a utility function to denote the availability of different versions. To constrain the number of replicas, we propose a probabilistic management policy and nodes to discard information with certain probability determined by the version of the information. Our objective is to find the best value of the probability to maximize the total utility value. Because new version is created with certain probability, nodes other than the source may not know whether the information stored in them is the latest version. Therefore, they can make decisions only according to the local state and decisions based on the local state can be seen as *local*-policy. We also explore the *global*-policy, that is, nodes understand the real state. We prove that the optimal policies in both cases conform to the *threshold* form. Simulations based on both synthetic and real motion traces show the accuracy of our theoretical model. Surprisingly, numerical results show that local-policy is better than the *global*-policy in some cases.

Keywords: Delay tolerant networks, stochastic updating information, probabilistic management policy, local-policy, global-policy, threshold form

1 INTRODUCTION

Different from traditional ad hoc networks, contemporaneous path between two nodes may not exist within given time in Delay Tolerant Networks (DTN) [1]. The concept of DTN is proposed to support many new emerging networking applications, where the end-to-end connectivity cannot be assumed due to various factors (e.g., sparse density, unpredictable node mobility, etc). Classic examples of DTN include deep-space exploration [2], mobile social network [3], vehicular network [4], etc.

In order to overcome the network partitions, nodes in DTN communicate through a *store-carry-forward* mode. Due to node mobility, different links come up and down. If the sequence of connectivity graphs over a time interval is overlapped, then an end-to-end path might exist, so the message should be *forwarded* through the existing link, *stored* and *carried* at the next hop until the next link comes up, and so on and so forth, until it reaches the destination [5]. At present, most of the works in DTN are oriented to the routing field, such as [5, 6, 7]. Beside routing, information access can be provided in various ways. For example, information can be pushed to users based on their interested profiles, such as the publish-subscribe systems [8]. Caching is another important way to provide information access service, and this is what we focus on in this paper. With the proliferation of powerful mobile devices, such as the iPhone, Blackberry, and Android devices, the potential applications of information caching in DTN are growing fast. Due to the characteristic of intermittent connection in DTN, caching policies often adopt the multi-replicas method. In essence, the multi-replicas policy exploits the storage capacity to increase the speed of information retrieval by replicating the message to many nodes rather than storing it in a central location that requires high processing power and represents a single point of failure. On the other hand, many routing methods also use the multi-replicas policy, such as [9, 10, 11]; but in most applications, nodes in DTN often work in the resource-constraint environment, for example, limited energy and buffer space, etc. Therefore, we think that the number of replicas cannot be too big. In this paper, we assume that the maximal number of replicas is limited, and the information is dynamic, that is, new versions can be created constantly. Dynamic information can be found in many situations, such as the files in weather forecast, software, etc. For dynamic information, older version is often less important than the newer one, and the best result is to keep all replicas in the network be the latest version. If we use a utility function to denote the availability of different versions, one main goal in studying of dynamic information is to maximize the total utility value. To describe the uncertain behavior in the creation of new versions, we assume that new versions are created with certain probability.

In this paper, we adopt discrete time model and the source creates new version in every time slot with given probability. The new versions are transmitted by the direct transmission method, so nodes can get the latest version only from the source, and the older version will be discarded directly when they get the latest one. Each node encountering with the source will get the new version, and then the node changes to state 1 (also indicates the state of information stored in this node). If the source created a new version in one slot, the state of all nodes carrying message which do not meet the source will be added by 1. Because the maximal number of replicas is limited, some versions need to be discarded. We use the probabilistic management policy. In particular, nodes discard information with certain probability determined by its state. In most applications, nodes may not know the real state, and hence they can make decision only according to the local state. This policy can be seen as *local-policy*. For comparison, we also study the *global-policy* in which we assume that every node knows its real state. We use discrete time Markov process to model the state evolving process and prove that the optimal policy in both cases conforms to the *threshold* form. The accuracy of our model was checked by some simulations. In addition, we give some numerical results of our model. These results show that the optimal policy really conforms to the *threshold* form, which shows the correctness of our theoretical result. Surprisingly, the results also show that *local-policy* may have better performance in some cases.

To our best knowledge, only the works in [12, 13] studied the optimal control problem of dynamic information in DTN; but these papers focused on the optimal forwarding policy which is different from our caching policy, that is, they study how to control the action of the source node to optimize their objective. In addition, they do not consider the limited buffer space. The rest of this paper is organized as follows. In the next section, we introduce some related works, and in Section 3 we give the network model. We study the evolving rule and the optimal management policy of the *local* and *global* policies in Sections 4 and 5, respectively. In Section 6, we explore the simulation and numerical results of our model. Finally, we conclude our main work.

2 RELATED WORK

Caching policy has been studied many years in the internet [14], mobile networks [15], etc. There are many excellent achievements in this field, but for simplicity, here we only introduce the works related to DTN.

Reich et al. proposed an optimal replication scheme in DTN in [16], which considered the impatience of content requesters. This work first proposed the question of optimizing caching policies to minimize certain utility function in a mobile network under the assumption of homogeneity, that is, mobile users meet each other with the same rate, storage capacities and so on. Optimization caching with heterogeneity nodes is proposed in [17]. Then the authors in [18] studied how to improve the data accessibility using WiFi Access Point. They proposed an optimal caching strategy so that the file retrieval probability within deadline is maximized, subject to storage constraint of each file server; but this paper did not consider the peer-to-peer data sharing among mobile users. Gao et al. proposed the cooperative caching policy with mobile users in DTN in [19], and their basic idea is to intentionally cache data at a set of Network Central Locations (NCLs), which can be easily accessed by other nodes in the network. Then the contact duration aware data replication method was proposed in [20, 21]. Now, some works have considered caching technology in more realistic environment, such as [22, 23].

All of the works above focused on the static information. In recent years, information that requires frequent updating is becoming much more common, and it can be seen as dynamic information. For dynamic information, a main goal is to minimize the average information age. Chaintreau et al. studied the distribution of information age using a spatial mean field method [24]. In that work, information was transmitted by gossiping protocol, but it did not consider the control problem. Altman et al. explored the aging controlling problem in [13]. They investigated the activation policy of mobile nodes based on the utility of their messages and studied the publishers' bonus strategy. Later, they studied how to maximize the availability of dynamic files with limited energy by controlling the probability of transmitting messages to relay nodes in [12]; but in some applications, buffer space is also a very important resource and may be limited, so the replicas number of one message cannot be too big. To our best knowledge, there are few works which studied how to maximize the total utility with limited number of replicas. In addition, all of the previous works solved their problem just by controlling the forwarding action of the source. In other words, the source forwards the information to others with certain probability. In fact, the information stored in the source is the latest version, and its utility value is much higher. Therefore, we think that the source should forward with probability 1 all the time, and other nodes discard the older version proactively to meet the constraint. For this reason, we propose the policy to control the action of the nodes other than the source, and this is different from state of the art works.

On the other hand, our caching policy is very similar to the buffer space management. The optimal buffer management policy in DTN was first proposed in [25], whose object was to minimize the average delivery delay or to maximize the average delivery rate based on global knowledge about the network, and then they proposed a joint scheduling and drop policy which can optimize different performance metrics in [26]; but these works only consider the static information. On the other hand, they mainly focus on which message should be discarded when the buffer space is full, so they are different from our work in essence.

3 NETWORK MODEL

Consider a network with a source node S and N relay nodes (labeled $1, 2, \dots, N$), so the total number of nodes is $N+1$. The density of the nodes is assumed to be sparse and they can communicate only when they move into the transmission range of each other, which means a communication contact. All nodes move independently in the network and their mobility models are assumed to be independent and identically distributed. In particular, we assume that the occurrence of contacts between two nodes follows a Poisson distribution, so the inter-meeting time follows the exponential distribution. This assumption has been used in wireless communications many years. At present, some works show that this assumption is only an approximation, and they reveal that nodes encounter with each other according to the power law distribution [27]. However, they also recognize that if you consider long traces, the tail of the distribution is not heavy-tailed, but exponential. In addition, the work in [28] shows that individual inter-meeting time can be shaped to be exponential by choosing an appropriate domain size with respect to a given time scale. Moreover, the work in [29] describes the inter-meeting time of individual human in mobile social networks by the exponential distribution and validates this model experimentally on real trace data. There are also some works which model the inter-meeting time between vehicles and find that it follows the exponential distribution by analyzing a large number of real car mobility traces [30, 31]. Therefore, our assumption is rational in some applications, and simulations based on both synthetic and real motion traces show that our theoretical model based on such assumption is very accurate.

In this paper, we adopt a discrete time model and assume that two nodes encounter with each other according to an exponential distribution with parameter λ. Considering time slot duration Δ, the t^{th} slot corresponds to interval $[t\Delta, (t+1)\Delta]$. Therefore, nodes meet each other with probability $1 - e^{-\lambda\Delta}$ in every slot and we set $q = 1 - e^{-\lambda\Delta}$. The source S creates messagge M at time 0, and then the new version of the message is created with probability u at every slot. We use the direct transmission routing method to transmit the updating information, that is, when S encounters with other nodes, it transmits the latest information to these nodes, but nodes other than the source cannot transmit with each other. In this paper, we assume that the bandwidth is big enough to transmit the information successfully in one contact. If a node carries the latest version we say that its state (or called age) is 1. Every time the source creates a new version, the state of those nodes that do not encounter with S will be added by 1. For example, node i was in state1 at $t1^{\text{th}}$ slot, if i did not encounter with S from $t1^{\text{th}}$ slot to the current time (t^{th} slot) and S created k new versions in this time duration, the state of node i becomes to be $k + 1$. By abuse of language, we assume that the state of information stored in one node equals to the node's state in the rest of this paper. Therefore, the state of one node can manifest the version of the information stored in it, and the bigger of the state value, the older of the version will be. Obviously, the state defined above is the real (or global) state. In most applications, because the source creates new version in every slot uncertainly, the relay nodes may not know its real state. Now, we propose the local state, which is defined as the number of slots that have elapsed since it last encountered with S. For example, node i met S at $t1^{\text{th}}$ slot, and it did not encounter with S in the following 3 slots. The local state of i at $(t1 + 3)^{\text{th}}$ slot will be 1+3. That is, node i assumes that the source creates new version certainly in every slot. Therefore, the local state is not smaller than the real state. In this paper, we also assume that when a node receives a new version it discards the older one immediately.

We use a utility function $U(k)$ to denote the availability of the information whose real state is k. It is easy to see that in many cases $U(k)$ is a decreasing function, that is, the utility value of the new version is bigger than the older one. When the age of the information is old enough, it is fully useless. In some cases, the outdated information may even cause bigger damage, so they should be discarded immediately. For simplicity, we assume that the version whose state is bigger than K (a constant) is useless and will be discarded directly. Because the buffer space may be limited, we think that the number of the replicas cannot be too big. Even if the buffer space is very big, much buffer space may be used to store data for other tasks. We also need to constrain the consumption of buffer space. In this paper, we consider the situation as follows: the buffer space is big, so the buffer cannot overflow. However, in order to make place for other tasks that may occur in the future, we need to constrain the buffer space for specific information. For example, mobile sensors are collecting data in certain environment. Because the data may be not transmitted to the sink timely, it should be stored in the sensors temporarily; but the new data may appear rapidly in the future, so even though all sensors have residual space, they may reserve the space for the new data. Therefore, the valid buffer space for the current data is limited and the replicas number is also limited. On the other hand, multi-replicas policy is often used to decrease the information retrieve speed, but limited copy number may already meet the demand, so too much replicas is not necessary. Therefore, our assumption is rational in some environments, and we assume that each information can have at most σ replicas in this paper. In addition, we assume that nodes are cooperative, and they are willing to receive the information whenever the source forwards towards them.

Let $X(k)$ denote the number of nodes (not including S) in state k and $E(X(k))$ denote its expectation. Our objective is to solve the optimization problem shown as follows:

$$\left\{\begin{array}{rcl} \mathrm{Max} M(\mathbf{p}) & = & \sum_{i=1}^{K} TU(i)E(X(i)) \\ \mathrm{Subject to} T(\mathbf{p}) & = & \sum_{i=1}^{K} E(X(i)) \leq \sigma. \end{array}\right. \tag{1}$$

Symbol $TU(i)$ denotes the utility value of information whose state is i (real or local state). When the real state is i, we have $TU(i) = U(i)$. When i is the local state, the value of $TU(i)$ can be obtained by Equation (17). Vector $\mathbf{p} = (p(1), p(2), \ldots, p(K))$ denotes the discarding probability in every valid state. For example, $p(k)$ denotes that information in state k will be discarded with probability $p(k)$. Therefore, we can call $\mathbf{p}$ as the control policy simply and this optimization problem turns into the finding of the optimal vector $\mathbf{p}$ to maximize the total utility value. Because the state is divided into local and global cases, $\mathbf{p}$ should also be calculated. We call them local-policy and global-policy. Later, we will explore the optimal management policy.

4 GLOBAL-POLICY

4.1 Evolving Process

For every relay node, there may be two events in every slot. The first event denotes that whether the relay nodes encounter with S, and the second event is that they need to decide whether to discard the information they carry. In this paper, we assume that the first event happens at the beginning of each slot, and then nodes will decide whether to discard the message. Though the first event is determined by the mobility model, which cannot be controlled by only one node, the relay nodes can control when to discard the message. Therefore, the assumption is reasonable. The evolving process can be described as a Markov process shown as follows:

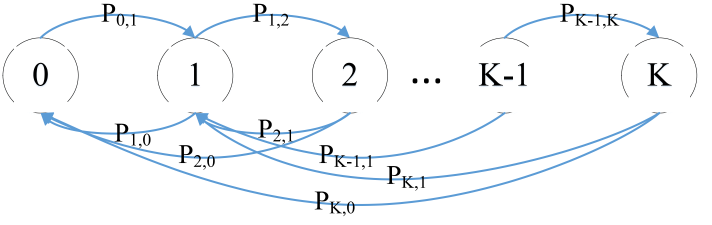


Figure 1. Local state transition graph

In Figure 1, symbol $P_{i,j}$ denotes the transition probability from state i to j. Nodes in state 0 do not have information at all. From the figure above, we can see that nodes in state 0 only can change to state 1, which happens when they encounter with S and they do not discard the message (with probability $q(1-p(1))$). When nodes meet S, they change to state 1 immediately, hence these nodes will discard the message with probability $p(1)$. Furthermore, because relay nodes encounter with S with probability q in one slot, we know that the total probability is $q(1-p(1))$. Nodes in state K may transmit to state 0 or 1. If they encounter with S and do not discard information (also with probability $q(1-p(1))$ as above case), they will move to state 1. However, if they discard information (with probability $qp(1)+(1-q)(1-u)p(K)$) they will go back to state 0. In addition, if they do not encounter with S and a new version is created (obviously with probability $u(1-q)$), they will also go back to state 0 because its state is bigger than K and it is useless. Therefore, the total probability of the transition from state K to 0 is $qp(1)+(1-q)(1-u)p(K)+u(1-q)$. Nodes in state i other than 0 and K may transmit to state 0, 1, or $i+1$. The first case happens when they discard M and the second case happens when they encounter with S and do not discard the message. The last case happens when they do not encounter with S and do not discard the message and the source created a new version. The probability of these events can become similar to that in state 0 and K, so we do not describe it again. From the analysis above, the transition probability is

$$\left\{\begin{array}{rcl} P_{i,i+1} & = & u(1-q)(1-p(i+1)), 0 < i < K \\ P_{i,1} & = & q(1-p(1)), i \geq 0, i \neq 1 \\ P_{i,i} & = & (1-q)(1-u)(1-p(i)), i > 1 \\ P_{1,1} & = & (q+(1-q)(1-u))(1-p(1)) \\ P_{i,0} & = & qp(1)+(1-u)(1-q)p(i)+u(1-q)p(i+1), 0 < i < K \\ P_{0,0} & = & qp(1)+(1-q) \\ P_{K,0} & = & qp(1)+u(1-q)+(1-a)(1-u)p(K). \end{array}\right. \quad (2)$$

Let π_i denote the stationary probability that a node is in state i, so we have

$$\left\{\begin{array}{rcl} \pi_i & = & \pi_{i-1}P_{i-1,i}+\pi_i P_{i,i}, i > 1 \\ \pi_1 & = & \sum_{i=0}^{K} \pi_i P_{i,1} \\ \sum_{i=0}^{K} \pi_i & = & 1. \end{array}\right. \quad (3)$$

Equation (3) defines linear system with $K+1$ equations and $K+1$ unknowns, that is to say, the system has a unique solution, and we have Conclusion 1 as follows:

Conclusion 1. The unique solution of Equation (3) is shown in Equation (4).

$$\left\{\begin{array}{rcl} \pi_1 & = & \frac{q(1-p(1))}{1-(1-q)(1-u)(1-p(1))} \\ \pi i & = & qu^{i-1}(1-q)^{i-1}\prod_{j=1}^{i}\frac{1-p(j)}{1-(1-q)(1-u)(1-p(j))}, i > 1 \\ \pi_0 & = & 1-\sum_{i=0}^{K}\pi_i. \end{array}\right. \quad (4)$$

Proof. First, from Equation (3) we can get

$$\pi_1 = \pi_1 P_{1,1} + \sum_{i=0\&i\neq 1}^{K} \pi_i P_{i,1} \Rightarrow \pi_1(1-P_{1,1}) = \sum_{i=0\&i\neq 1}^{K} \pi_i P_{i,1}.$$

From Equation (1), we can see that each state beside state 1 changes to state 1 with the same probability, so we have

$$\pi_1(1-P_{1,1}) = \sum_{i=0\&i\neq 1}^{K} \pi_i P_{i,1} = \sum_{i=0\&i\neq 1}^{K} \pi_i q(1-p(1)) = q(1-p(1))(1-\pi_1).$$

Furthermore, combining with the equation about $P_{1,1}$ shown in Equation (2), we can get the solution of π_1. From Equation (3), we know that, for $i > 1$,

$$\begin{aligned}\pi_i &= \pi_{i-1}P_{i-1,i}+\pi_i P_{i,i}\\ \Rightarrow \frac{\pi_i}{\pi_{i-1}} &= \frac{P_{i-1,i}}{1-P_{i,i}}\\ \Rightarrow \frac{\pi_i}{\pi_{i-1}}\frac{\pi_{i-1}}{\pi_{i-2}}\cdots\frac{\pi_2}{\pi_1} &= \frac{P_{i-1,i}}{1-P_{i,i}}\frac{P_{i-2,i-1}}{1-P_{i-1,i-1}}\cdots\frac{P_{1,2}}{1-P_{2,2}}\\ \Rightarrow \pi_i &= \pi_1\frac{P_{i-1,i}}{1-P_{i,i}}\frac{P_{i-2,i-1}}{1-P_{i-1,i-1}}\cdots\frac{P_{1,2}}{1-P_{2,2}}.\end{aligned}$$

Combining with the equations about $P_{i,i}$ and $P_{i,i+1}$ shown in Equation (2), we can get the second part of Equation (4). Therefore, Conclusion 1 is correct. □

In the stable state, the expression of $X(i)$ is

$$X(i)=\sum_{j=1}^{N}\delta_j(i). \tag{5}$$

Symbol $\delta_j(i)$ denotes the event that node j is in state i, so we have $p(\delta_j(i))=\pi_i$. The expectation of $X(i)$ can be described as follows:

$$E(X(i))=E\left(\sum_{j=1}^{N}\delta_j(i)\right)=N\pi_i. \tag{6}$$

Therefore, the constraint function in Equation (1) can be expressed as

$$\begin{aligned}T(\mathbf{p}) &= \sum_{i=1}^{K}E(X(i))\\ &= Nq\sum_{i=1}^{K}u^{i-1}(1-q)^{i-1}\prod_{j=1}^{i}\frac{1-p(j)}{1-(1-q)(1-u)(1-p(j))}.\end{aligned} \tag{7}$$

Let $x(j)=1-(1-q)(1-u)(1-p(j))$, $j=1,2,\ldots,K$. Equation (7) will be converted to the next expression,

$$\begin{aligned}T(\mathbf{x}) &= Nq\sum_{i=1}^{K}u^{i-1}(1-q)^{i-1}\prod_{j=1}^{i}\frac{1-p(j)}{1-(1-q)(1-u)(1-p(j))}\\ &= Nq\sum_{i=1}^{K}u^{i-1}(1-q)^{i-1}\prod_{j=1}^{i}\frac{1-x(j)}{(1-q)(1-u)x(j)}\\ &= \frac{Nq}{1-q}\sum_{i=1}^{K}u^{i-1}(1-u)^{-i}\prod_{j=1}^{i}(\frac{1}{x(j)}-1).\end{aligned} \tag{8}$$

For simplicity, we further let $y(j) = 1/x(j) - 1, j = 1, 2, \ldots, K$, so we have,

$$T(\mathbf{y}) = \frac{Nq}{1-q} \sum_{i=1}^{K} u^{i-1}(1-u)^{-i} \prod_{j=1}^{i} y(j). \tag{9}$$

Obviously, the objective function $M(\mathbf{p})$ can be converted to $M(y)$ shown as follows:

$$M(\mathbf{y}) = \frac{Nq}{1-q} \sum_{i=1}^{K} U(i) u^{i-1}(1-u)^{-i} \prod_{j=1}^{i} y(j). \tag{10}$$

4.2 Optimal Management Policy

Because $0 \leq p \leq 1, 1 \leq i \leq K$, we have, $1-(1-q)(1-u) \leq x(j) \leq 1$, $j = 1, 2, \cdots, K$. Therefore, we can get the value of $y(j)$ easily, that is, $0 \leq y(j) \leq (1-q)(1-u)/(1-(1-q)(1-u))$.

First, we consider the static probability as a special case, that is, $y(i) = y, 1 \leq i \leq K$. In this case, the objective and constraint functions can be described as

$$\begin{cases} M(\mathbf{y}) &= \frac{Nq}{1-q} \sum_{i=1}^{K} U(i) u^{i-1}(1-u)^{-i} y^i \\ T(\mathbf{y}) &= \frac{Nq}{1-q} \sum_{i=1}^{K} u^{i-1}(1-u)^{-i} y^i. \end{cases} \tag{11}$$

From the above formula, we can easily see that both the objective and constraint functions increase with y. Therefore, there exists the unique solution to maximize $M(\mathbf{p})$ and this can be got by Equation (11) easily. Now we start to explore the dynamic policy and we will see that the optimal policy conforms to the threshold form, which is defined as follows:

Definition 1. Suppose there are two constants $0 \leq z \leq (1-q)(1-u)/(1-(1-q)(1-u))$, $1 \leq h \leq K$ and they satisfy: $y(i) = (1-q)(1-u)/(1-(1-q)(1-u))$, $0 < i < h$; $y(i) = z$, $i = h$; $y(i) = 0$, $i > h$. We call the policy threshold policy and denote it by $y(h, z)$.

Obviously, $y(h, z)$ may have three forms:

Case 1: $y(h, z) = (z, 0, \ldots, 0)$, $h = 1$;

Case 2: $y(h, z) = ((1-q)(1-u)/(1-(1-q)(1-u)), \ldots, (1-q)(1-u)/(1-(1-q)(1-u)), z)$, $h = K$;

Case 3: $y(h, z) = ((1-q)(1-u)/(1-(1-q)(1-u)), \ldots, (1-q)(1-u)/(1-(1-q)(1-u)), z, 0, \ldots, 0)$, $1 < h < K$;

According to the definition of stochastic order [32], $y(h, z)$ is increasing with h. We have the following theorem:

Lemma 1. Non-threshold policy cannot be optimal.

Proof. Given a policy y, if we can find another policy v and it satisfies $T(v) = T(y)$ and $M(v) > M(y)$, we can say that y is not optimal. Suppose y is a non-threshold policy, we can get a constant c, and it satisfies: $y(c) < (1-q)(1-u)/(1-(1-q)(1-u))$, $y(c+1) > 0$. Here we assume that $y(i) > 0$, $i < c$, or nodes cannot go to state c. We will show that we can always find two positive constants ε_1 and ε_2 and set policy v as: $v(c) = y(c) + \varepsilon_1 < (1-q)(1-u)/(1-(1-q)(1-u))$, $v(c+1) = y(c+1) - \varepsilon_2 > 0$, $v(i) = y(i)$, $i \neq c, c+1$, and we have $T(v) = T(y)$. Let $\theta = v(c)v(c+1) - y(c)y(c+1)$; according to Equation (9), we can get,

$$T(\mathbf{v}) - T(\mathbf{y}) = \frac{Nq}{1-q}\sum_{i=1}^{K} u^{i-1}(1-u)^{-i}(\prod_{j=1}^{i} v(j) - \prod_{j=1}^{i} y(j)) = 0.$$

The formula shown above equals to

$$\begin{aligned}\sum_{i=c}^{K} u^{i-1}(1-u)^{-i}(\prod_{j=1}^{i} v(j) - \prod_{j=1}^{i} y(j)) &= y(1)y(2)\ldots y(c-1)u^{c-1}(1-u)^{-c} \\ &\quad \left(\varepsilon_1 + \frac{u}{1-u}\theta + \left(\frac{u}{1-u}\right)^2 \theta y(c+2)\ldots \right. \\ &\quad \left. + \left(\frac{u}{1-u}\right)^{K-c} \theta y(c+2)\ldots y(K)\right) \\ &= 0.\end{aligned}$$

Therefore, we can have

$$\begin{cases} \varepsilon_1 + D\theta = 0 \\ D = \frac{u}{1-u} + (\frac{u}{1-u})^2 y(c+2)\ldots + \left(\frac{u}{1-u}\right)^{K-c} y(c+2)\ldots y(K). \end{cases}$$

Obviously, we have $\theta < 0$ and we can get

$$\varepsilon_2 = \frac{\varepsilon_1(Dy(c+1)+1)}{D(y(c)+\varepsilon_1)}.$$

We can get the proper value for ε_1 and ε_2 satisfying the above equation easily. Now we will prove that $M(v) > M(y)$. Obviously, the following equation is correct.

$$\frac{M(\mathbf{v}) - M(\mathbf{y})}{Nqu^{c-1}(1-u)^{-c}(y(1)y(2)\ldots y(c-1))/(1-q)}$$

$$
\begin{aligned}
&= \varepsilon_1 U(c) + \theta \left(\frac{u}{1-u} U(c+1) + \left(\frac{u}{1-u}\right)^2 U(c+2) y(c+2) + \ldots \right. \\
&\quad \left. + \left(\frac{u}{1-u}\right)^{K-c} U(K) y(c+2) \ldots y(K) \right) \\
&= U(c) \left(\varepsilon_1 + \theta \left(\frac{u}{1-u} + \left(\frac{u}{1-u}\right)^2 y(c+2) + \ldots \right.\right. \\
&\quad \left.\left. + \left(\frac{u}{1-u}\right)^{K-c} y(c+2) \ldots y(K) \right)\right) \\
&\quad + \theta \left(\frac{u}{1-u} (U(c+1) - U(c)) + \left(\frac{u}{1-u}\right)^2 (U(c+2) - U(c)) y(c+2) \right. \\
&\quad \left. + \ldots + \left(\frac{u}{1-u}\right)^{K-c} (U(K) - U(c)) y(c+2) \ldots y(K) \right) \\
&= U(c)(\varepsilon_1 + \theta D) + \theta \left(\frac{u}{1-u} (U(c+1) - U(c)) \right. \\
&\quad + \left(\frac{u}{1-u}\right)^2 (U(c+2) - U(c)) y(c+2) \\
&\quad \left. + \ldots + \left(\frac{u}{1-u}\right)^{K-c} (U(K) - U(c)) y(c+2) \ldots y(K) \right) \\
&= \theta \left(\frac{u}{1-u} (U(c+1) - U(c)) + \left(\frac{u}{1-u}\right)^2 (U(c+2) - U(c)) y(c+2) \right. \\
&\quad \left. + \ldots + \left(\frac{u}{1-u}\right)^{K-c} U(K) y(c+2) \ldots y(K) \right)
\end{aligned}
$$

Because $\theta < 0$ and the utility function is decreasing, we know that $M(v) > M(y)$. □

From Equations (8) and (9), we can see that both the objective and constraint function is increasing with h; if we fix h, they also increasing with z. Therefore, we have

$$\max(T(\mathbf{y}(h,z))) = T(\mathbf{y}(K, (1-q)(1-u)/(1-(1-q)(1-u)))). \tag{12}$$

If $\max(T(y(h,z))) < \sigma$, the constraint is useless, so we do not need to discard any message. In the next part of this paper, we only consider the case $\max(T(y(h,z))) \geq \sigma$.

From Theorem 1 we know that if the optimal policy exists, it must be of threshold form. Now, we will explore whether the optimal policy exists and how to get it. We can easily solve these questions by Theorem 2.

Lemma 2. The optimal threshold policy exists and it saturate the constraint in Equation (8).

Proof. Given $\sigma > 0$, we know that $T(y(1,0)) = 0 < \sigma$, so there exist some feasible solutions. The optimal policy can be got by following algorithm:

1. Set $h = 1$, $z = (1-q)(1-u)/(1-(1-q)(1-u))$. The corresponding policy is $y = (z, 0, \dots, 0)$. If we have $T(y(h,z)) > \sigma$, because $T(y(h,z))$ is increasing with h, we can say that the optimal policy is $(b, 0, \dots, 0), 0 \leq b \leq (1-q)(1-u)/(1-(1-q)(1-u))$. The real value of b can be got by $T(y(1,b)) = \sigma$ easily. If $T(y(h,z)) < \sigma$, we can get better policy through increasing the value of h and carry out step (2).
2. Let $h = h+1$, $z = (1-q)(1-u)/(1-(1-q)(1-u))$. If we have $T(y(h,z)) \geq \sigma$, we know that the optimal policy satisfies: $y(i) = (1-q)(1-u)/(1-(1-q)(1-u))$, $0 < i < h$; $y(i) = b$, $i = h$; $y(i) = 0$, $i > h$. Now, the value of h is known, and can get b by $T(y(h,b)) = \sigma$. If $T(y(h,z)) < \sigma$, we carry out step (2) again till $h = K$.

Combining with Theorem 1, the policy got through the above algorithm is optimal and it saturates the constraint in Equation (1). □

5 LOCAL-POLICY

5.1 Evolving Process

In the local-policy case, because the relay nodes do not know their real state, they cannot judge whether the information stored in them is useless. For simplicity, we assume that when the local state is bigger than K, they discard the information directly. Obviously, they may discard some versions which are younger than K, that is, they may discard the information which is still useful. The evolving process can be described as a Markov process which is the same as Figure 1. However, the number in Figure 1 denotes the local state now. From the figure, we can see that nodes in state 0 can only transmit to state 1 in the next slot which happens when they encounter with S and they do not discard the message. Nodes in state K may change to state 0 or 1. If they encounter with S and do not discard M, they will move to state 1, if they discard M or do not encounter with S, they will go to state 0. Nodes in state i other than 0 and K may change to state 0, 1, or $i+1$. The first case happens when they discard M and the second case happens when they encounter with S and do not discard the message. The last case happens when they do not encounter with S and do not discard the message (whether new version is created is not related). Therefore, the transition probability is

$$\begin{cases} P_{i,i+1} &= (1-q)(1-p(i+1)), 0 < i < K \\ P_{i,1} &= q(1-p(1)), i \geq 0 \\ P_{i,0} &= qp(1) + (1-q)p(i+1), 0 < i < K \\ P_{K,0} &= P_{0,0} = qp(1) + 1 - q. \end{cases} \tag{13}$$

Note that when nodes are in state 1, they do not change their state in the next slot only when they meet S and do not discard the message. If they do not encounter with S, they will assume that a new version is created, and their local state will be added by 1. Therefore, the transition probability from state 1 is not the same as that in the global case.

The stationary distribution is shown as follows:

$$\left\{\begin{array}{rcl} \pi_i & = & \pi_{i-1}P_{i-1,i}, i > 1 \\ \pi_1 & = & \sum_{i=0}^{K} \pi_i P_{i,1} \\ \sum_{i=0}^{K} \pi_i & = & 1. \end{array}\right. \tag{14}$$

From Equations (13) and (14), we have

$$\pi_i = q(1-q)^{i-1} \prod_{j=1}^{i} (1-p(j)), i \geq 1. \tag{15}$$

In fact, the state in this section is not the nodes' real state, because they do not know whether the source really created a new version in one slot. If the node's local state is $k(>0)$, this means that there are $k-1$ slots elapsed since it last encountered with S, and its real state may be 1, 2, …, k. If the source created $i < k$ new versions in the k slots, the node's real state is $i+1$ and this probability is denoted as $P_k(i+1)$ which is shown as follows:

$$P_k(i+1) = C_{k-1}^{i} u^i (1-u)^{k-i-1}. \tag{16}$$

Therefore, we can obtain the utility value when the local state is k, which is shown as follows:

$$TU(k) = \sum_{i=0}^{k-1} P_k(i+1)U(i+1). \tag{17}$$

Furthermore, Equation (1) can be converted to the expression as follows:

$$\left\{\begin{array}{rcl} M(\mathbf{p}) & = & \sum_{i=1}^{K} E(X(i)) \sum_{k=1}^{i} C_{i-1}^{k-1} u^{k-1}(1-u)^{i-k} U(k) \\ & = & Nq \sum_{i=1}^{K} (1-q)^{i-1} \prod_{j=1}^{i} (1-p(j)) \sum_{k=1}^{i} P_i(k)U(k) \\ T(\mathbf{p}) & = & \sum_{i=1}^{K} E(X(i)) = Nq \sum_{i=1}^{K} (1-q)^{i-1} \prod_{j=1}^{i} (1-p(j)). \end{array}\right. \tag{18}$$

5.2 Optimal Management Policy

Similar to the global-policy, we also first consider the static policy as a special case, that is, $p(i) = p$, p is a constant and we have $0 \leq p \leq 1$, $1 \leq i \leq K$. Therefore, Equation (18) can be described as

$$\left\{\begin{array}{rcl} M(\mathbf{p}) & = & Nq \sum_{i=1}^{K} (1-q)^{i-1}(1-p)^i \sum_{k=1}^{i} P_i(k)U(k) \\ T(\mathbf{p}) & = & \sum_{i=1}^{K} E(X(i)) = Nq \sum_{i=1}^{K} (1-q)^{i-1}(1-p)^i \leq \sigma. \end{array}\right. \tag{19}$$

From the above formula, we can easily see that the objective and constraint function both decrease with p. Therefore, there exists unique solution to maximize $M(p)$ and this can be got by Equation (19) easily.

To get the optimal dynamic probability, let $y(i) = 1 - p(i)$, $i = 1, 2, \dots, K$. Therefore, Equation (18) can be described as

$$\begin{cases} M(\mathbf{p}) &= Nq\sum_{i=1}^{K}(1-q)^{i-1}\prod_{j=1}^{i}y(j)\sum_{k=1}^{i}P_i(k)U(k) \\ T(\mathbf{p}) &= \sum_{i=1}^{K}E(X(i)) = Nq\sum_{i=1}^{K}(1-q)^{i-1}\prod_{j=1}^{i}y(j). \end{cases} \tag{20}$$

We can see that they have the same structure as Equations (9) and (10), and Theorems 1 and 2 are correct in the *local*-policy, too.

6 SIMULATION AND NUMERICAL RESULTS

6.1 Simulation Result

In this section, we will evaluate the accuracy of our theoretical model, and we run several simulations using the Opportunistic Network Environment (ONE) simulator [33], which is a typical platform for DTN simulation. The simulation is based on both synthetic mobility model and real-world-based scenarios. The synthetic model is Random Waypoint (RWP) mobility model, which is commonly used in many mobile wireless networks. In this model, the simulation terrain is $1\,000\,\mathrm{m} \times 1\,000\,\mathrm{m}$, and the nodes' speed varies from 0.5 to 1.25 m/s. The communication range of every node is 5 m. For the real-world-based scenario, we use the Poisson contact model. Specially, the Poisson process with $\lambda = 3.71 \times 10^{-6}s^{-1}$ is used to generate node contact events. This value is obtained from the vehicle model, which is based on real motion traces from about 2 100 operational taxis for about one month in Shanghai city collected by GPS [34, 35]. Authors in [31] proposed a least-fitting method to identify the exponential parameter and find that the inter-contact time is well proper. Finally, we carry out the simulation based on the Infocom '05 datasets, which includes 41 nodes [36].

Because nodes encounter with each other according to the exponential distribution, we can get the meeting probability q in one slot, which is shown as follows

$$q = 1 - e^{-\lambda\Delta}. \tag{21}$$

Other settings are: $N = 100$, $\sigma = 20$, $K = 10$, $U(i) = 1/i$, $i = 1, 2, \dots, K$. Since our goal is to check the accuracy of the theoretical model, we only carry out the simulation with optimal policy which is the threshold form. In this section, we increase u from 0 to 1 continuously, and carry out the simulation when $\Delta = 100\,\mathrm{s}$ and $1\,000\,\mathrm{s}$, respectively. Simulation results are shown in Figures 2 and 3. For the Infocom '05 datasets, we have $N = 41$ and other settings remain unchanged. Then, we can get Figure 4.

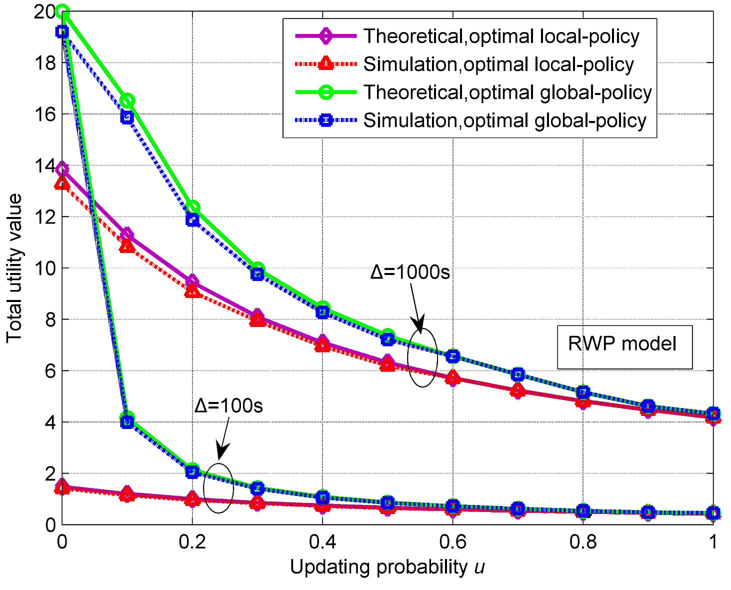


Figure 2. Simulation and theoretical results comparison with RWP mobility model

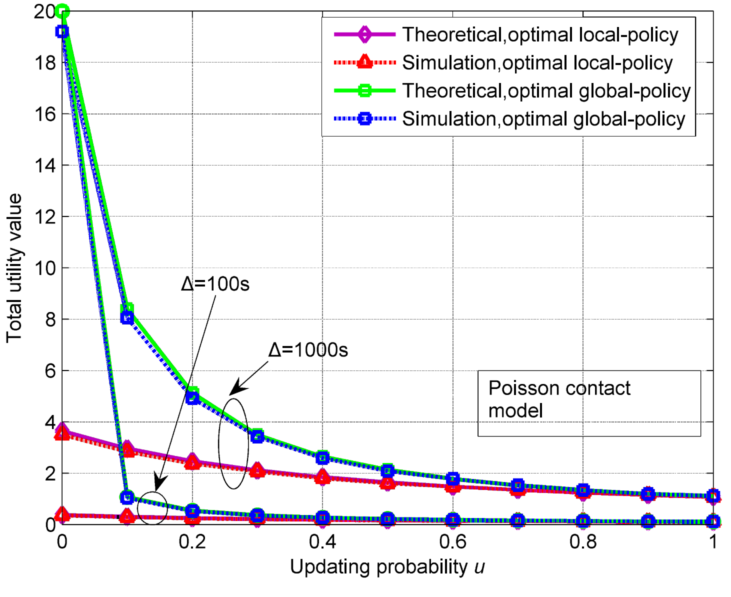


Figure 3. Simulation and theoretical results comparison with Poisson contact model

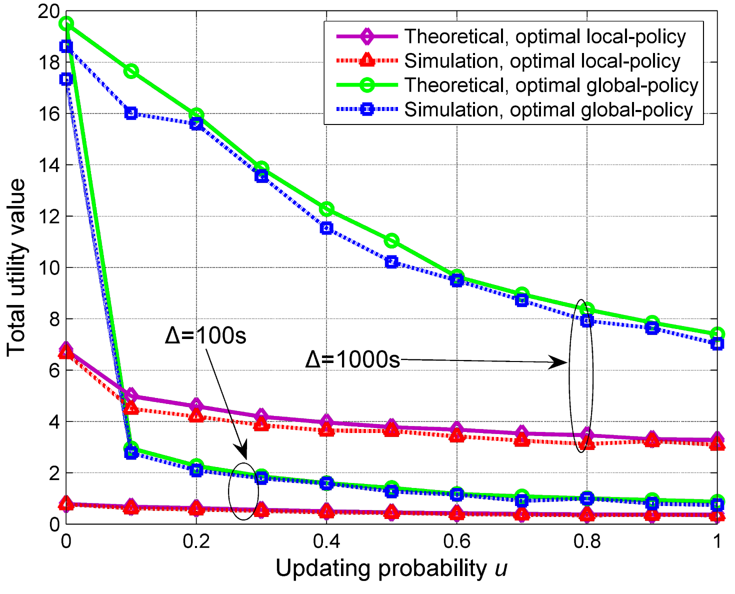


Figure 4. Simulation and theoretical results comparison with Infocom 2005 datasets

Comparing the simulation and theoretical results, we can see that our model is very accurate, and the average deviation between the simulation and theoretical results is very small. For example, the deviation is about 3.92 % for the RWP mobility model and 2.1 % for the Poisson mobility model. The result in Figure 4 shows that the deviation between the simulation and numerical results is bigger than that in Figures 2 and 3. In fact, some works have shown that the Infocom '05 datasets may conform to the power law and exponential decay distribution [27], so if we use the exponential model to fit this datasets, it will bring bigger deviation. However, the deviation is only about 7.612 %, and we think that it is not too big. Therefore, our model has more applications. For this reason, we can use the theoretical results obtained by our model in the performance evaluation in the next subsection.

Moreover, from these results we also can see that if the updating probability u is bigger, the total utility value may be smaller in both cases. This means that if the source creates new versions frequently, users cannot obtain these new versions timely. In the next section, we will further evaluate the performance of the system in different cases by numerical results.

6.2 Performance Analysis with Numberical Results

From Figures 2 and 3, we also can see that if the time interval Δ is bigger, the total utility value is bigger, too. This is because that when Δ is bigger, the value of q

shown in Equation (21) will be bigger, hence the relay nodes have bigger probability to encounter with the source S. Therefore, with fixed updating probability u, the information version in them may be younger. This result shows that the encounter probability q may have certain impact on the performance. Now, we begin to explore this problem further with numerical result. The basic setting is: $N = 100$, $\sigma = 50$, $K = 5$, $u = 0.3$, $U(i) = 1/i$, $i = 1, 2, \ldots, K$. We give the total utility value with optimal policy, which is the *threshold* form. In addition, we also give the result with optimal *static* probability. The result is shown in Figure 5.

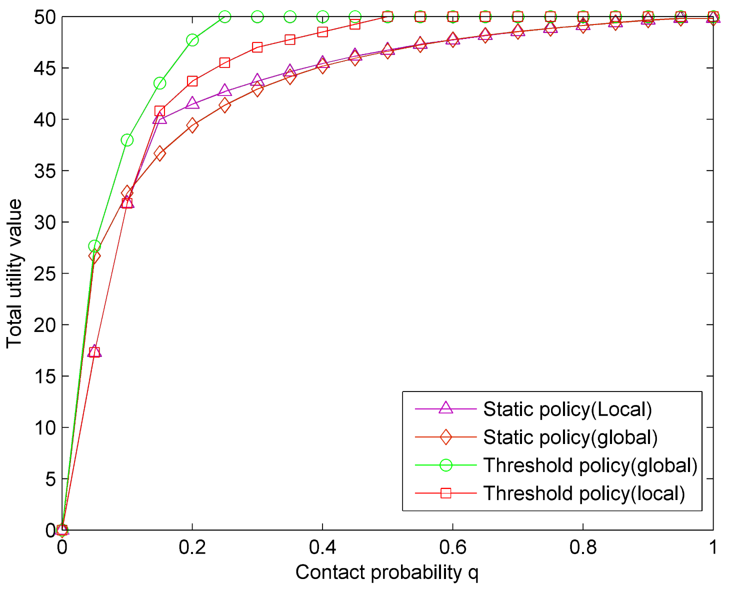


Figure 5. Impact of contact probability q

From Figure 5, we can see that the *threshold global*-policy is the best one. Surprisingly, when $0.1 < q < 0.4$, the *static global*-policy is worse than the *static local*-policy. This is because when a node's local state is bigger than K, its real state may be smaller than K, the *global*-policy know this, so this policy will treat those nodes the same as those whose local state is smaller than K, but the *local*-policy will discard these versions directly. It is easy to see that if a node's local state is bigger than the other's, the real state of the former is also bigger than that of the latter. Therefore, the *local*-policy can discard the older versions proactively, so the newer versions may be stored with bigger probability and the total utility value is bigger, too. For example, there are 80 nodes carrying message whose real state is smaller than K. To meet the constraint, nodes with static *global*-policy will discard the message with probability $1 - /80 = 1 - 50/80 = 0.375$. If there are 30 nodes in above 80 nodes whose local state is bigger than K, the *static local*-policy will discard these versions directly. Therefore, there are $80 - 30 = 50$ nodes carrying message, and nodes with *static local*-policy will discard message with probability $1 - \sigma/50 = 1 - 50/50 = 0$. When $q < 0.1$, the relay nodes encounter with S with small probability; in this case, the number of replicas is small, so even if we do not discard any information whose real state is not bigger than K, the constraint may not be violated. However, the *local*-policy does not know the real state, it will discard the versions whose local state is bigger than K (including the versions whose real state is smaller than K), so in this situation, its performance is worse than the *global*-policy. When $q > 0.4$, relay node will have more chances to meet with the source S, so most of the nodes' state is smaller than K, obviously, the *local* and *global* policies will have little difference which is proved by Figure 5.

As described above, information whose state is bigger than K is useless. Therefore, if the value of K is smaller, there may be more outdated versions in the network and this parameter may have some impact on the network performance. To explore this problem, we let $q = 0.05$ and increase K from 5 to 50. Other settings are the same as those in Figure 5. Numerical result can be found in Figure 6.

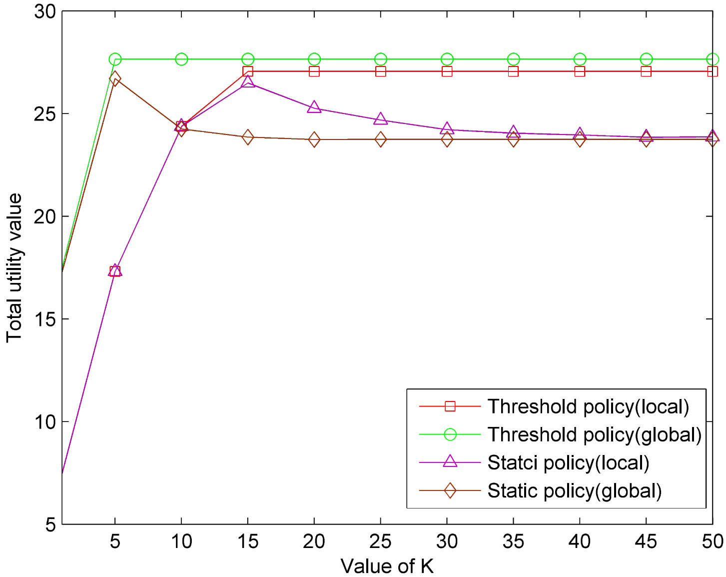


Figure 6. Impact of K

As shown in Figure 6, the performance of the *threshold* policies in both cases is increasing with K and this is because when K is bigger, less information will be discarded. In addition, the *threshold* policies will get their maximal value soon and then keep invariance. This is because some versions have to be discarded, or the constraint condition in Equation (1) will be violated. Through the definition of the *threshold* policy, we know that older versions will be discarded, so we cannot improve the performance by increasing the value of K again. From the figure, we also can see that *threshold global*-policy is the best one. The performance of the *static* policies in Figure 6 first increases with K and then decreases; at last, they keep invariance, too. When K is small, the number of the replicas whose state is smaller than K cannot saturate the constraint condition, so if we increase the value of K the performance will be improved. However, when K is bigger, we have to discard some versions, or it will violate the constraint. Because the static policy treats all versions whose state is smaller than K equally, if we increase the value of K, we have bigger probability to discard the newer versions, so the total utility value will decrease. When K is big enough, state in nearly all of the versions is smaller than K, so the nodes do not need to discard information at all and the performance will keep invariance.

Parameter σ is an important component in the constraint condition in Equation (1), so we need to explore its impact on the performance of our model. In this section we set $q = 0.05$ and σ is increased from 10 to 80, other settings are also the same as those in Figure 5. The result is shown in Figure 7.

Figure 7 shows that when reaching to about 25, the *global*-policies both dynamic and static are better than the *local*-policy, because when σ is bigger, there is no need to discard any information, but the *local*-policy will discard all versions whose local state is bigger than K but their real state is smaller than K, that is, they will discard some useful information, so they have worse performance. However, when σ is smaller than 25, the *static global*-policy is the worst one. This is because in this case many versions have to be discarded. The *local*-policy will first discard the versions whose local state is bigger than K, but these versions' real state may be smaller than K, so the *static global*-policy will treat them the same as those whose state is smaller. Obviously, compared to the *local*-policy, the *static global*-policy has bigger probability to discard the newer versions, so its performance is the worst one. On the other hand, though the optimal *static local*-policy may be better than the *static global*-policy in some cases, the optimal policy is the *threshold global*-policy all the time. In addition, the result in Figure 7 shows that when the total number of replicas is small (e.g., < 10), the deviation between the *global*-policy and *local*-policy is very small.

Symbol u is the probability that the new version is created in every slot. Here, we let $N = 100$, $\sigma = 50$, $q = 0.05$, $U(i) = 1/i$, $i = 1, 2, \ldots, K$ and increase u from 0 to 1 continuously. In this subsection, we set $K = 5$, 40, and explore the results in both cases, respectively. The result in Figure 8 shows that the *threshold global*-policy is better in both cases and the performance of all policies decreases with u. When u is bigger, new versions are created more frequently, but the basic settings remain unchanged and the updating information is transmitted with the same speed, so the availability of the information decreases more quickly, and the total utility value decreases with u, obviously. Numerical result in this section also shows that when

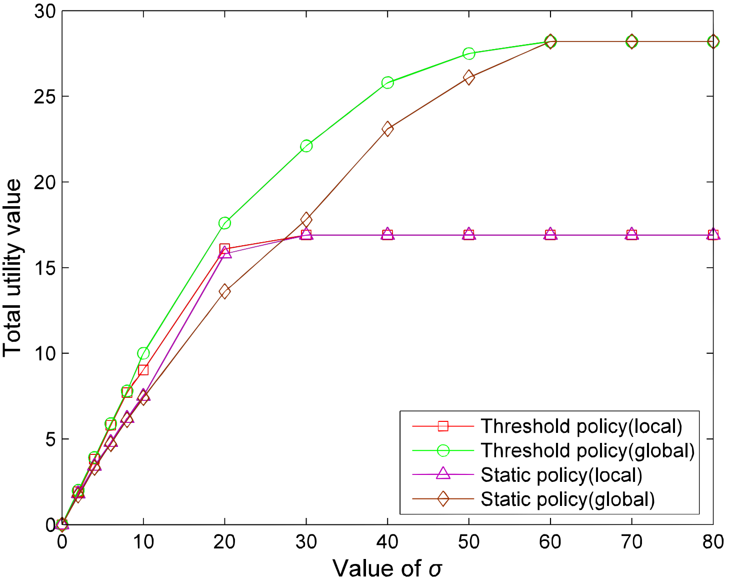


Figure 7. Impact of σ

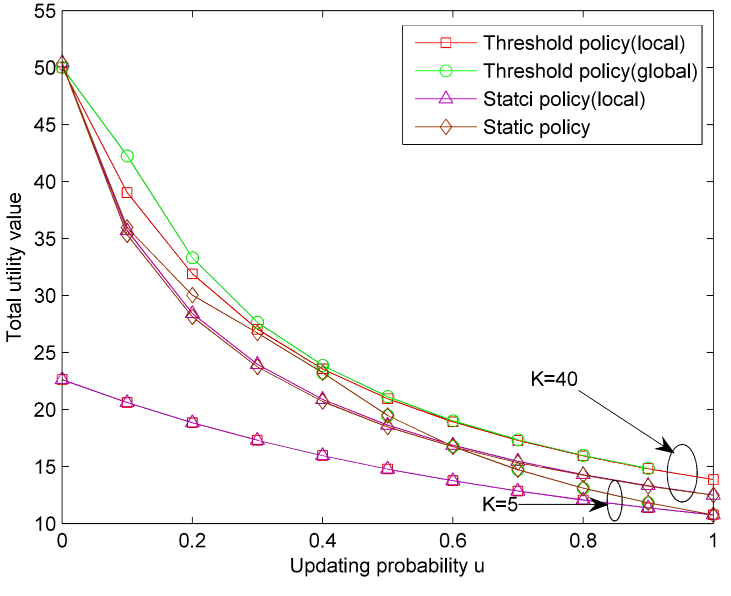


Figure 8. Impact of u

K is smaller, *global*-policy is better than the *local*-policy, but when K is bigger, the *static global*-policy is not better than the *local*-policy which is consistent with the result above.

Numerical results in the above sections all assumed that the utility function is $U(i) = 1/i, i = 1, 2, \ldots, K$; in fact, any positive deceasing function can be used. Therefore, the number of feasible utility function is infinite and we cannot study every one of them. In fact, the function $U(i) = 1/i$ is a special case of the power function: $U(i) = a^{*}t^{-b}$, $a, b > 0$, $i = 1, 2, \ldots, K$. For simplicity, we use the general power function as our utility function and explore the performance of our model. In this section, the basic settings are: $N = 100$, $\sigma = 50$, $K = 20$, $q = 0.05$, $u = 0.3$. Because we have proved that the optimal policy in both *global*-policy and *local*-policy conforms to the *threshold* form, we only give the numerical results of the *threshold* policy shown as Figure 9. Obviously, the optimal *global*-policy is better.

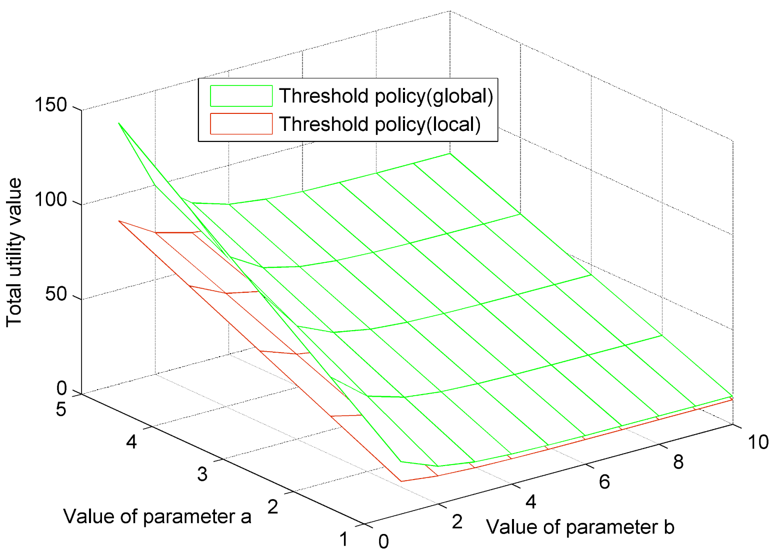


Figure 9. Performance with general power function

From all of the numerical results above, we can see that the *threshold global*-policy is the best one all the time. If we cannot get the global knowledge, we should select the *threshold local*-policy. However, to get the *threshold* policy, we also need some parameters about the network, and this may be hard in some applications. In this situation, nodes have to discard information with equal probability and this is the *static* policy. In addition, our numerical result shows that *static local*-policy is better than the *static global*-policy in some environments. Therefore, the *local*-policy has certain advantage, too.

7 CONCLUSIONS

In this paper, we study how to maximize the total utility value in DTN with stochastic updating dynamic information. To meet the constraint condition, information in the relay nodes will be discarded with certain probability, which is decided by the information state. We use the discrete time model and the source node creates new version in every slot with probability u. Nodes with the latest version are in state 1, every time the source creates a new version, state of versions stored in the relay nodes which do not encounter with the source will be added by 1. Because the new version is created uncertainly, nodes other than the source may not know their real state, so they have to make decision with local information and we call the policy *local*-policy. We also explore the *global*-policy in this paper. We study the evolving process using Markov process and prove that the optimal policy in both cases conform to the *threshold* form. Finally, we give the simulation and numerical results. Surprisingly, if we use the *static* policy, *local*-policy may be better than the *global*-policy in some cases.

We assume that the updating information can be transmitted in one slot successfully in this paper. In fact, nearly all of the theoretical models in DTN have similar hypothesis [12, 13], but in many situations, the message may be very big, so they cannot be transmitted in only one slot. In the future, we want to relax the hypothesis and explore the Markov process with different message size. Note that this paper focuses on single message case and does not consider the wireless contention. Our model can be extended to the case with multiple messages easily when there is no wireless contention. When the wireless contention exists, we have to consider the schedule policy of the buffer space, the model will be more complex and this will be our future work, too. In addition, based on our model, how to design the proper query method to access the information efficiently is annother important problem.

Acknowledgement

The research reported in this paper was supported by NSFC No. 61401482.

REFERENCES

[1] Fall, K.: A Delay-Tolerant Network Architecture for Challenged Internets. Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM 2003), ACM, New York 2003, pp. 27–34, DOI 10.1145/863955.863960.

[2] Papastergiou, G.—Psaras, I.—Tsaoussidis, V.: Deep-Space Transport Protocol: A Novel Transport Scheme for Space DTNs. Computer Communications, Special Issue on Delay and Disruption-Tolerant Networks, Vol. 32, 2009, pp. 1757–1767.

[3] Gao, W.—Cao, G.: On Exploiting Transient Contact Patterns for Data Forwarding in Delay Tolerant Networks. Proceedings of the 18^{th} IEEE International Conference on Network Protocols (ICNP), 2010, pp. 193–202.

[4] Chen, T.—Zhu, L.—Wu, F.—Zhong, S.: Stimulating Cooperation in Vehicular Ad Hoc Networks: A Coalitional Game Theoretic Approach. IEEE Transactions on Vehicular Technology, Vol. 60, 2011, No. 2, pp. 566–579.

[5] Spyropoulos, T.—Psounis, K.—Raghavendra, C. S.: Efficient Routing in Intermittently Connected Mobile Networks: The Single-Copy Case. IEEE/ACM Transactions on Networking (ToN), Vol. 16, 2008, No. 1, pp. 63–76.

[6] Whitbeck, J.—Conan, V.: HYMAD: Hybrid DTN-MANET Routing for Dense and Highly Dynamic Wireless Networks. Computer Communications, Vol. 33, 2010, No. 13, pp. 1483–1492.

[7] Yuan, Q.—Cardei, I.—Wu, J.: Predict and Relay: An Efficient Routing in Disruption-Tolerant Networks. Proceedings of 10^{th} ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), May 2009, pp. 95–104.

[8] Costa, P.—Mascolo, C.—Musolesi, M.—Picco, G. P.: Socially-Aware Routing for Publish-Subscribe in Delay-Tolerant Mobile Ad Hoc Networks. IEEE Journal on Selected Areas in Communications, Vol. 26, 2008, No. 5, pp. 748–760.

[9] Spyropoulos, T.—Psounis, K.—Raghavendra, C. S.: Efficient Routing in Intermittently Connected Mobile Networks: The Multiple-Copy Case. IEEE/ACM Transactions on Networking (ToN), Vol. 16, 2008, No. 1, pp. 77–90.

[10] Bulut, E.—Wang, Z.—Szymanski, B.: Cost Effective Multi-Period Spraying for Routing in Delay Tolerant Networks. IEEE/ACM Transactions on Networking (ToN), Vol. 18, 2010, No. 5, pp. 1530–1543.

[11] Liu, C.—Wu, J.—Cardei, I.: Message Forwarding in Cyclic MobiSpace: The Multi-Copy Case. Proceedings of the 6^{th} IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS '09), Macau SAR, China, 2009, pp. 413–422.

[12] Altman, E.—Nain, P.—Bermond, J.-C.: Distributed Storage Management of Evolving Files in Delay Tolerant Ad Hoc Networks. Proceedings of 2009 IEEE INFOCOM, Rio de Janeiro, Brazil, 2009, pp. 1431–1439.

[13] Altman, E.—El-Azouzi, R.—Menasche, D. S.—Xu, Y.: Forever Young: Aging Control in DTNs, `http://arxiv.org/PS_cache/arxiv/pdf/1009/1009.4733v1.pdf`.

[14] Podlipnig, S.—Böszörmenyi, L.: A Survey of Web Cache Replacement Strategies. ACM Computer Surveys, Vol. 35, 2003, No. 4, pp. 374–398.

[15] Yang, K.—Chiu, G.: A Hybrid Pull-Based with Piggybacked Push Protocol for Cache Sharing. The Computer Journal, Vol. 54, 2011, No. 12, pp. 2017–2032.

[16] Reich, J.—Chaintreau, A.: The Age of Impatience: Optimal Replication Schemas for Opportunistic Networks. Proceedings of the 5^{th} ACM International Conference on Emerging Networking Experiments and Technologies (CoNEXT), 2009, pp. 85–96.

[17] Ioannidis, S.—Massoulie, L.—Chaintreau, A.: Distributed Caching Over Heterogeneous Mobile Networks. Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, 2010, pp. 311–322.

[18] Huang, Y.—Gao, Y.—Nahrstedt, K.—He, W.: Optimizing File Retrieval in Delay-Tolerant Content Distribution Community. Proceedings of the 29th Conference on Distributed Computing Systems, 2009, pp. 308–316.

[19] Gao, W.—Cao, G.—Iyengar, A.—Srivatsa, M.: Supporting Cooperative Caching in Disruption Tolerant Networks. Proceedings of the 31st International Conference on Distributed Computing Systems (ICDCS), 2011, pp. 151–161, DOI: 10.1109/ICDCS.2011.76.

[20] Zhu, X.—Li, Q.—Cao, G.—Dai, Y.—Szymanski, B.—La Porta, T.: Social-Based Cooperative Caching in DTNs: A Contact Duration Aware Approach. Proceedings of the 8th IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS), 2011, pp. 92–101.

[21] Zhu, X.—Li, Q.—Gao, W.—Dai, Y.: Contact Duration Aware Data Replication in Delay Tolerant Networks. Proceedings of the 19th IEEE International Conference on Network Protocols (ICNP), 2011, pp. 236–245.

[22] Hyytiä, E.—Virtamo, J.—Lassila, P.—Kangasharju, J.—Ott, J.: When Does Content Float? Characterizing Availability of Anchored Information in Opportunistic Content Sharing. Proceedings of the 30th IEEE International Conference on Computer Communications (INFOCOM), Shanghai, China, 2011, pp. 3137–3145.

[23] Ott, J.—Hyytiä, E.—Lassila, P.—Vaegs, T.—Kangasharju, J.: Floating Content: Information Sharing in Urban Areas. Proceedings of the 30th IEEE International Conference on Computer Communications (INFOCOM), Shanghai, China, 2011, pp. 136–146.

[24] Chaintreau, A.—Le Boudec, J.-Y.—Ristanovic, N.: The Age of Gossip: Spatial Mean Field Regime. Proceedings of the ACM International Conference on Measurement and Modeling of Computer Systems, 2009, pp. 109–120, DOI: 10.1145/1555349.1555363.

[25] Krifa, A.—Barakat, C.—Spyropoulos, T.: Optimal Buffer Management Policies for Delay Tolerant Networks. 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON '08), San Francisco, CA, USA, 2008, pp. 260–268, DOI: 10.1109/SAHCN.2008.40.

[26] Krifa, A.—Barakat, C.—Spyropoulos, T.: Message Drop and Scheduling in DTNs: Theory and Practice. Technical Report, HAL INRIA, 2010, `http://hal.archives-ouvertes.fr/docs/00/54/23/09/PDF/TMC-DTN.pdf`.

[27] Karagiannis, V.—Le Boudec, J.-Y.—Vojnovič, M.: Power Law and Exponential Decay of Intercontact Times between Mobile Devices. Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking (MobiCom '07), ACM, New York, NY, USA, 2007, pp. 183–194.

[28] Cai, H.—Eun, D.: Crossing Over the Bounded Domain: From Exponential to Power-Law Intermeeting Time in Mobile Ad Hoc Networks. IEEE/ACM Transactions on Networking (ToN), Vol. 17, 2009, Iss. 5, pp. 1578–1591.

[29] Gao, W.—Li, Q.—Zhao, B.—Cao, G.: Multicasting in Delay Tolerant Networks: A Social Network Perspective. Proceedings of the 10th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), 2009, pp. 299–308, DOI: 10.1145/1530748.1530790.

[30] Lee, K.—Yi, Y.—Jeong, J.—Won, H.—Rhee, I.—Chong, S.: Max-Contribution: On Optimal Resource Allocation in Delay Tolerant Networks. Proceedings of the 29th IEEE International Conference on Computer Communications (INFOCOM), San Diego, CA, USA, 2010, pp. 1–9.

[31] Zhu, H.—Fu, L.—Xue, G.—Zhu, Y.—Li, M.—Ni, L. M.: Recognizing Exponential Inter-Contact Time in VANETs. Proceedings of the 29th IEEE International Conference on Computer Communications (INFOCOM), San Diego, CA, USA, 2010, pp. 101–105.

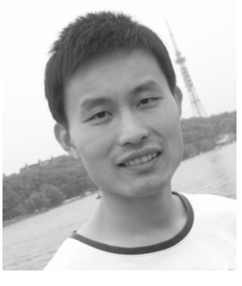
[32] Shaked, M.—Shantikumar, J. G.: Stochastic Orders and Their Applications. Academic Press, New York 1994.

[33] Keranen, A.—Ott, J.—Karkkainen, T.: The One Simulator for DTN Protocol Evaluation. Proceedings of the 2nd International Conference on Simulation Tools and Techniques (Simutools '09), Article No. 55, ISBN: 978-963-9799-45-5, DOI 10.4108/ICST.SIMUTOOLS2009.5674.

[34] Li, Y.—Su, G.—Wu, D. O.—Jin, D.—Su, L.—Zeng, L.: The Impact of Node Selfishness on Multicasting in Delay Tolerant Networks. IEEE Transactions on Vehicular Technology, Vol. 60, 2011, No. 5, pp. 2224–2238.

[35] S. J. U. Traffic Information Grid Team, Grid Computing Center, Shanghai Taxi Trace Data [Online], Available on `http://wirelesslab.sjtu.edu.cn/`.

[36] Hui, P.—Chaintreau, A.—Scott, J.—Gass, R.—Crowcroft, J.—Diot, C.: Pocket Switched Networks and Human Mobility in Conference Environments. Proceedings of the 2005 ACM SIGCOMM Workshop on Delay-Tolerant Networking (WDTN '05), ACM New York, NY, USA 2005, pp. 244–251, ISBN: 1-59593-026-4, DOI 10.1145/1080139.1080142.



Yahui Wu received the B. Sc. and M. Sc. degrees in information system engineering from National University of Defense Technology (NUDT), Changsha, China, in 2006 and 2008, respectively. Then he received his Ph. D. degree from NUDT in 2013. He is currently a lecturer in the Science and Technology on Information Systems Engineering Laboratory of NUDT. His research fields include mobile delay tolerant networks, wireless communications, and performance evaluation.

Su Deng received the B. Sc. degree from Naval University of Engineering, Wuhan, China, in 1983 and his Ph. D. degree from National University of Defense Technology, in 2001. He is currently a Professor in Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology. His research fields include mobile delay tolerant networks, wireless communications, and information management.

Hongbin Huang received the B. Sc. and Ph. D. degrees in information system engineering from National University of Defense Technology, Changsha, China, in 1995 and 2006, respectively. He is currently an Associate Professor in the Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology. His research fields include peer-to-peer networks, mobile sensor networks, and performance evaluation.