Augmenting MANET's File Availability by Simulation and Resource Allotment for Effective File Sharing

P. V. Mathalai Dharshini[1] Maria Jesula Sebasty. S[2] S.Vimal[3]
[1, 2, 3]Department of Information Technology
[1, 2, 3]Jeppiaar Engineering College, Chennai, India.

Abstract—*Currently in mobile ad hoc networks (MANETs) the file sharing applications have drawn attention over the years. The diacritic properties of MANETs like node mobility, Limited communication range and resource allotment declines the efficiency of file querying. A novel method to ease this catch is to create file duplicates in the network with minimum average querying detain. Particularly current file simulation protocols in MANETs have few drawbacks. Foremost the average querying detain is high because limited resources are not allotted properly to the files. Secondary, the chance of determining file availability is low because MANETs consider storage as a means for duplicates hence the frequency of file possessors to contact other nodes become very less. Higher availability of files is only possible when a node has higher contact frequency with other nodes. This becomes clearer in distributed MANETs, where nodes contact disruptively. Here in this paper we propose a novel approach for file simulation by considering both node storage and nodes contact frequency. To achieve the proposed rule a protocol is designed by which the average querying detain is minimized. Therefore this proposed protocol can achieve shorter average querying detain at a lower cost than current simulation protocols.*

Keywords—MANETs, Peer-to-Peer, File Sharing, File Availability

I. Introduction

Mobile devices i.e., Smartphone's and laptops, being an emerging technology, it envisaged the future of MANETs participation in mobile devices. The erstwhile has a relatively dense node distribution in a local area whereas the hindmost has scarcely distributed nodes that rarely contact each other.[1] The mobile file sharing applications prompts the analysis on the peer-to-peer (P2P) file sharing over such MANETs[2].

In general MANETs are referred as Normal MANETs and Disconnected MANETs(or detain tolerant networks (DTNs)).The confined P2P model has three advantages. Foremost, file sharing can be done without the use of a base station (e.g., rural area). Next, overloaded server's problem in current client-server based file sharing systems can be avoided with P2P architecture. At last, the wasted peer to peer communication opportunities are utilized among mobile nodes. As a result, nodes can freely and unassertively access and share files in the distributed MANET environment.

However, the diacritic properties of MANETs like node mobility, Limited communication range and resource allotment have extended the difficulties in realizing a P2P file sharing system[3]. An effective way to enhance file availability and reduce file querying detain is File simulation.

It creates duplications for a file to improve its probability of being confronted by requests. But, it is unrealistic to make all the nodes hold the duplication of all files considering the limited node resources. Also, file querying detain is always a main cover in a file sharing system. Users often desire to receive their requested files rapidly no matter whether the files are in demand or not. Thus, a critical issue is raised for further investigation on how to allot the limited resource in the network to different files for duplication so that the overall average file querying detain is minimized.

A number of file simulation protocols have been proposed for normal MANETs lately[4]. In these protocols frequently accessed files are duplicated at each individual node, or a group of nodes create one copy for each file they frequently query. In distributed MANETs for efficient data retrieval or message routing, they primarily follow an intuitive way in which frequently queried data are cached and placed on mobile nodes that are visited frequently.

In spite of the efforts, current file simulation protocols lack a rule to allot limited resource to different files for copy creation in order to achieve the minimum overall mean querying detain. In this paper, we introduce a new concept of resource allotment for file simulation, which considers both node storage and node contact frequency.

II. Related Work

A. Sharing Files in normal MANETs

File sharing in MANETs, is based on simulation of files and is decided by the individual or group of nodes according to file meeting frequency.

Now three file simulation protocols are being used [5]: Static Access Frequency (SAF), Dynamic Access Frequency and Neighborhood (DAFN) and Dynamic Connectivity based Grouping (DCG). In SAF, each node duplicates its frequently visited files until its available storage is used up.SAF may lead to creation of many duplicate files if the neighboring nodes have the same interested files. DAFN removes duplicate copies among neighbors. Among a group of nodes with frequent connections the duplicate copies are reduced further by DCG.

B. Sharing Files in Disconnected MANETs

Currently many techniques are being used to share files in Disconnected MANETs. In a more general P2P scenario, all mobile nodes are both file servers and clients. Gao et al [6] proposed a cooperative caching method in DTNs (Detain Tolerant Network) by copying each file to the node in each network central repository or node, which is frequently visited by other nodes. When the central node is full, less-popular simulations are moved to its neighbor nodes. However, these methods fail to consider that the mobility of a node affects the availability of files or messages and further optimize the simulation distribution to enhance file availability or routing success rate.

C. Symbolizing optimal simulation issue

Along with file duplication the general operation to model the expected file querying detain is presented. Let $\widehat{p_n}$ be the probability that a node's recently met node in the time interim is node n, then, the mean number of time interim needed to contact a particular file, say file f, can be represented as:

$$\overline{\overline{\mathbf{T}}}_{\mathbf{f}} = \frac{\mathbf{1}}{\sum_{\mathbf{n=1}}^{\mathbf{N}} \widehat{\mathbf{p}}_{\mathbf{n}} \mathbf{X}_{\mathbf{nf}}} \quad (\mathbf{1})$$

Then, the mean number of interims needed to assure a request is

$$\overline{\overline{\mathbf{D}}} = \sum_{\mathbf{f=1}}^{\mathbf{F}} \mathbf{q}_{\mathbf{f}} \overline{\overline{\mathbf{T}}}_{\mathbf{f}} = \sum_{\mathbf{f=1}}^{\mathbf{F}} \frac{\mathbf{q}_{\mathbf{f}}}{\sum_{\mathbf{n=1}}^{\mathbf{N}} \widehat{\mathbf{p}_{\mathbf{n}}} \mathbf{X}_{\mathbf{nf}}} \quad (\mathbf{2})$$

With equation (2), we can formulate the overall maximization issue by minimizing $\overline{\widehat{D}}$, which can be further used to reduce the superlative simulation rule. However, the calculation of $\widehat{p_n}$ may be difficult and makes the minimization problem significant.

III. THEORETICAL ANALYSIS OF OVERALL SUPERLATIVE FILE SIMULATION

A. Node Motion Models

In the research area of MANETs, for the normal MANETs the [7] Random Way Point model (RWP) is used and for the disconnected MANETs the [8] Community-Based Mobility Model is used. Thus, we also use the two models to represent the two types of MANETs in theoretical analysis.

1) Random Waypoint Model for Normal MANETs

In RWP, nodes frequently move to a irregularly selected point at a random speed, (i.e.) each node can roughly contact other nodes. However, nodes usually have different probability of contact frequency with other nodes in reality. We hence let each node have a randomly obtained speed, rather than continuously varying a node's speed in different paths.

2) Community-Based Mobility Model for Disconnected MANETs

In some routing algorithms of disconnected MANETs, to describe the node mobility, this model has been used. The entire test area is separated into different sub-areas called caves. One community is detained by each cave and node belongs to one or more community called home community. When moving, a node has probability P_{in} to wait in the home community and probability $1 - P_{in}$ to visit a foreign community and it mostly moves within its home community.

3) Presumptions and restrictions

The two mobility models discussed lies on two Presumptions: 1) the probability of contacting a particular node is identical for all nodes (RWP model) or for all nodes is in its home community (community-based model) and 2) nodes move unaided in the network (both models). These two presumptions may not hold in real cases, which limit the applicability in different real scenarios.

B. Conceptual Study

In this section, we conceptually study the effect of the file duplication distribution on the total query efficiency in MANETs under the two models. Refer to Table 1 for the meanings of notations.

Notations	Description
q_f	The probability of querying file f in the system
p_n	The probability that the next encountered node is node n
p_f	The probability of obtaining file f in the next encountered node
N	Total number of nodes
V_n	Node $n's$ contacting ability (i.e. frequency of meeting nodes)
S_n	Storage space of node m
A	Average contacting ability of all nodes in the system
F	Total number of files in the system
S_f	Size of file f
X_{nf}	Whether node n contains file f or not
V_{fl}	contacting ability of the l^{th} node that holds file f
n_f	The number of nodes holding file f or its duplicates
A_f	Allot resource for file f for simulation
T_f	Average number of time intervals needed to contact file f
P_f	Priority value of file f, $P_f = \sqrt{q_f/S_f}$
R	Total amount of resource in the system
T	Average number of time intervals needed to contact a file

Table. 1: Notations

1) Superlative File Simulation with the RWP model

In the RWP model, we can assume that exponential distribution is followed among the nodes inter-contacting Time. Then, the probability of contacting a node is individualistic with the previous encountered node. Therefore, we define the **contacting ability** of a node as the mean number of nodes it contacts in a unit time and use it to enquire into the superlative file simulation. Specifically, if a node is able to contact more nodes, it has higher probability of being experienced by other nodes. That is

$$\mathbf{p}_{\mathbf{n}} = \frac{\mathbf{V}_{\mathbf{n}}}{\sum_{\mathbf{l=1}}^{\mathbf{N}} \mathbf{V}_{\mathbf{l}}} = \frac{\mathbf{V}_{\mathbf{n}}}{\mathbf{V}_{\mathbf{N}}} \quad (\mathbf{3})$$

We use vector $(V_{1f}, V_{2f}, \ldots, V_{nf})$ to denote the assembling abilities of a group of nodes holding file f or its copies, Then, the probability that a node obtains its requested file f is,

$$\mathbf{p}_{\mathbf{f}} = \sum_{\mathbf{n=1}}^{\mathbf{N}} \mathbf{p}_{\mathbf{n}} \mathbf{X}_{\mathbf{nf}} = \sum_{\mathbf{n=1}}^{\mathbf{N}} \frac{\mathbf{V}_{\mathbf{n}} \mathbf{X}_{\mathbf{nf}}}{\mathbf{AN}} = \sum_{\mathbf{l=1}}^{\mathbf{nf}} \frac{\mathbf{V}_{\mathbf{fl}}}{\mathbf{AN}} \quad (\mathbf{4})$$

A node's probability of being confronted by other nodes is proportionate to the contact ability of the node. This indicates that files present in nodes with higher contacting ability have higher availability than files in nodes with lower contacting ability. When a copy is created in a node, it engages the memory on the node. This means that the copies naturally consume both the storage resource and the contacting ability resource of the node. Then, the total amount of resource in the system (R) is:

$$\mathbf{R} = \sum_{\mathbf{n=1}}^{\mathbf{N}} \mathbf{S}_{\mathbf{n}} \mathbf{V}_{\mathbf{n}} \quad (\mathbf{5})$$

Thus, the Total Resource assigned to file f is:

$$\mathbf{R_f = S_f \sum_{l=1}^{nf} V_{fl}} \quad (6)$$

Based on Equation (6), Equation (4) can be represented as

$$\mathbf{p_f = \frac{s_f \sum_{l=1}^{nf} V_{fl}}{s_f AN}} \quad (7)$$

Thus, the possibility of contacting file f after l $l = (1,2,3,\dots)$ time interims is $(1-p_f)^{l-1}p_f$ and the number of time interims needed for a node to contact a node containing file f is

$$\mathbf{T_f = \sum_{l=1}^{\infty} l\,(1-p_f)^{l-1}p_f = \frac{1}{p_f} = \frac{S_f AN}{R_f}} \quad (8)$$

We use $q_f \in [0; 1]$ to denote the probability of a node's deriving a request for file f in the system during a unit of time period. Then, the mean number of interims needed to please a request is

$$\mathbf{T = \sum_{f=1}^{F} q_f T_f = \sum_{f=1}^{F} AN \frac{q_f S_f}{R_f} = AN \sum_{f=1}^{F} \frac{q_f S_f}{R_f}} \quad (9)$$

We aim to reduce the overall file querying detain (i.e., D) by file simulation. Thus, the problem of superlative resource allotment is then converted to finding the superlative amount of resource (R_f) for each file j under the restriction of total available resource in order to achieve the minimum average querying detain. The problem of superlative resource allotment is expressed by

$$\mathbf{\min(D) = \min\left\{\sum_{f=1}^{F} \frac{q_f S_f}{R_f}\right\} = \min\left\{\sum_{f=1}^{F} \frac{Q_f}{R_f}\right\}} \quad (10)$$

Subject to: $\sum_{f=1}^{F} R_f \le R$

Equation (9) also indicates that each R_f should be as large as possible in order to minimize T. Therefore, we assume all resources (R) are assigned.

$$\mathbf{\sum_{f=1}^{F} R_f = R} \quad (11)$$

By applying Formula (11), Formula (10) is changed to

$$\mathbf{\min(D) = \min\left\{\frac{Q_1}{R_1} + \dots + \frac{Q_f}{R - (R_1 + R_2 + \dots + R_{f-1})}\right\}} \quad (12)$$

Next, we try to find the value of R_f that satisfies Formula (12). Specifically, we first calculate the first order (necessary) condition by differentiating T on each R_f $(1 \le f \le F-1)$ respectively, and find the value of R_f that makes the modified formula equal to 0. The resultant formulas after differentiation are

$$\mathbf{\frac{Q_1}{R_1^2} - \frac{Q_f}{\{R - (R_1 + R_2 + \dots + R_{F-1})\}} = 0} \quad (13)$$

$$\mathbf{\frac{Q_{F-1}}{R_{F-1}^2} - \frac{Q_F}{\{R - (R_1 + R_2 + \dots + R_{F-1})\}^2} = 0} \quad (14)$$

Combine all of the above F - 1 equation, we get

$$\mathbf{\frac{Q_1}{R_1^2} = \frac{Q_2}{R_2^2} = \frac{Q_3}{R_3^2} = \dots = \frac{Q_{F-1}}{R_{F-1}^2} = \frac{Q_F}{R_F^2}} \quad (15)$$

To attain the least mean detain, the second order (sufficient) condition should be greater than 0 as below:

$$\mathbf{\frac{-2Q_1}{R_1^3} - \frac{-2Q_F}{\{R - (R_1 + R_2 + \dots + R_{F-1})\}^3} > 0} \quad (16)$$

$$\mathbf{\frac{-2Q_{F-1}}{R_{F-1}^3} - \frac{-2Q_F}{\{R - (R_1 + R_2 + \dots + R_{F-1})\}^3} > 0} \quad (17)$$

If Equation (15) is true, based on Equation (11), Formulas (16) and (17) can be altered to below.

$$\mathbf{\left(\frac{1}{R_F} - \frac{1}{R_1}\right)\frac{2Q_1}{R_1^2} > 0} \quad (18)$$

$$\mathbf{\left(\frac{1}{R_F} - \frac{1}{R_{F-1}}\right)\frac{2Q_{F-1}}{R_{F-1}^2} > 0} \quad (19)$$

When $R_F < R_f$ $(f \in [1, F-1]$, Equations (18) and (19) conditions are satisfied. Remember that above result is obtained when we replace R_F with $R - (R_1 + R_2 + \dots + R_{f-1})$ in Equation (10). If we replace R_l (l ∈ [1,F]) with $R - (R_1 + .. + R_{l-1} + R_{l+1} + R_F)$,the second order is also satisfied when $R_l < R_f$ (f ∈ [1, F]; f ≠ l). In summary, the second sequence is satisfied when the resource assigned for each file is less than the resource assigned for any other file. This condition is always true because there always exists a file with the lowest assigned resource. Therefore, as long as the first sequence condition (Equation (15)) is satisfied, the second sequence condition is also satisfied.

Then, according to Equation (11) and Equation (15), we can see that the superlative assignment is

$$\mathbf{R_f = \frac{\sqrt{Q_j}}{\sum_{l=1}^{F}\sqrt{Q_l}} R(f = 1, 2, 3, \dots, F)} \quad (20)$$

This means that the superlative resource allotment is obtained through the square root scheme, i.e., the part of resource for file f is in direct portion of the square root of S_f

$$\mathbf{R_f \propto \sqrt{Q_j} => S_f \sum_{l=1}^{nf} V_{fl} \infty \sqrt{S_f q_f}} \quad (21)$$

That is,

$$\mathbf{\sum_{l=1}^{nf} V_{nl} \infty \sqrt{\frac{q_f}{S_f}} => \sum_{l=1}^{nf} V_{fl} \infty P_f} \quad (22)$$

We call $\sqrt{\frac{q_f}{s_f}}$ the Priority Value (P) of file f

Based on Formula (22), we derive the Superlative File Simulation Rule (SFSR) that gives the direction for the superlative resource allotment for each file that leads to the least average file querying detain under the RWP model.

SFSR-In order to achieve least global file querying detains, the sum of the contacting ability of copies nodes of file f should be proportional to P_f.

2) Superlative File Simulation with the Community Based Mobility model

In this model rather than considering contacting ability, we consider each node's **performing ability**. It is defined as a node's capability to please queries in the system (denoted by V_n) and is calculated based on the node's capacity to please queries in each community.

We use N_g to denote the number of nodes in community g. Then, community g holds $\frac{N_g}{N}$ fragment of nodes in the system. Node n's performing capability to community g depends on both the number of different nodes in g it contacts in a unit time period (denoted by M_{ng}), and the number of queries produced by nodes in g. In this model, since nodes' file interests are stable during a certain time period, we presume that each node's querying design remains stable.

As a result to please queries from a small community, a file occupant has low capability. Thus, we combine each community's fraction of nodes (i.e., $\frac{N_g}{N}$) into the computation of the performing ability. Therefore,

$$\mathbf{A_n \sum_{g=1}^{G} M_{ng} \frac{N_g}{N}} \qquad (23)$$

Where G is the total number of communities. Given n_f nodes that hold file f or its copies, we again use vector$(V'_{f1}, V'_{f2}, \ldots, V'_{fl}, \ldots, V'_{nn_f})$ to denote the performing ability of these nodes. Then, the overall capability of nodes in the system to satisfy requests for file f (denoted by o_f) is the total of all the performing ability times a redundancy removal factor ∞.

$$\mathbf{O_f = \infty \sum_{l=1}^{Af} V'_n (\infty \in [0,1])} \qquad (24)$$

∞ denotes the "deduction" on the overall performing ability considering the fact that the performing abilities of dissimilar file possessors may overlay.

Then, the number of time interims needed to please a request for file f is

$$\mathbf{T'_f = \frac{1}{O_f} = \frac{1}{\infty \sum_{l=1}^{nf} V'_{nl}}} \qquad (25)$$

Similar to Equation (6), the total resource assigned to file f can be represented by

$$R'_f = S_f \sum_{l=1}^{nf} V'_{nl}$$

As an outcome, the mean number of time interims needed to satisfy a request in the system is

$$\mathbf{D' = \sum_{f=1}^{F} q_f T'_f = \sum_{f=1}^{F} q_f \frac{1}{\infty \sum_{l=1}^{nf} V'_{fl}} = \frac{1}{\infty} \sum_{f=1}^{F} \frac{q_f S_f}{R'_f}} \qquad (26)$$

Then, the problem of superlative resource assignment can be expressed by

$$\mathbf{\min(D') = \min\left\{\sum_{f=1}^{F} \frac{q_f S_f}{R'_f}\right\} = \min\left\{\sum_{f=1}^{F} \frac{Q_f}{R'_j}\right\}} \qquad (27)$$

We can find that Equation (27) is the same as Equation (10). Then, we follow the same process after Equation (10) and deduce the SFSR rule in disconnected MANETs as

$$\mathbf{\sum_{l=1}^{nf} V'_{fl} \infty \sqrt{\frac{q_f}{S_f}} => \sum_{l=1}^{nl} V_{fl} \infty P_f} \qquad (28)$$

IV. Dispersed File Simulation Protocol

In this section, a dispersed file simulation protocol that can approximately realize the superlative file simulation rule (SFSR) with the two mobility models in a dispersed method is proposed. At first the challenges to understand the SFSR and way to overcome them are introduced.

A. Challenges and Solutions to obtain the SFSR

Challenge 1: Allotting resources without a main server

SFSR shows that in order to realize the overall superlative searching detain, each file's popularity (q_f), size (S_f), and the system resource (R) information must be known for simulation creation. Especially, in case there are F files in the system with $S_1 q_1 \ldots S_f q_f$ and total resource R, the resource Allotted to file f (R_f) should be

$$\mathbf{R_f = R \times \sqrt{S_f q_f} / \sum_{l=1}^{F} \sqrt{S_l q_l}} \qquad (29)$$

A way to achieve this is to setup a main server to access all information, perform resource allotment, and spread the information to file owners to duplicate their files. But the nature of distributed network becomes an obstacle here.

Solution to Challenge 1: Resource Competition

SFSR (i.e. Formula (22)) requires that for each file, the sum of its duplicate nodes contacting abilities, $\sum_{l=1}^{nF} V_{Fl}$, is proportional to its priority value P. In other words, SFSR can be shown by

$$\mathbf{P_1 / \sum_{l=1}^{n_1} V_{1l} = P_2 / \sum_{l=2}^{n_2} V_{2l} \ldots = P_F / \sum_{l=1}^{nF} V_{Fl}} \qquad (30)$$

Where n_j ($j \in [1,2, \ldots F]$) represents the number of duplicate nodes of file j. Then, we can let each file, say file j, regularly compete for the resource with its current $P_j / \sum_{l=1}^{nj} V_{Fl}$.Thus, files with larger $P_j / \sum_{l=1}^{nj} V_{Fl}$ wins more competitions and receive more resource and files with smaller $P_j / \sum_{l=1}^{nj} V_{Fl}$ only win few competitions and receive less resource. By enabling file owners to distributively compete for resource for their files, we can realize OFRR without a central server.

Challenge 2: Competition for Dispersed Resources

In a MANET, all unoccupied resource is spread among multiple nodes in the network which are in motion. It leads to three complications. Foremost, dissimilar file possessors are spread and can barely assemble together to organize the resource competition. Next, after a file is duplicated to an amount of nodes, it is hard to gather the popularity of the copies to upgrade the P of the file. Third, since the amount of nodes met by a file possessor is restricted, a sole file possessor cannot disperse copies productively and rapidly.

Solution to Challenge 2: Dispersive Competition On Particular Resources.

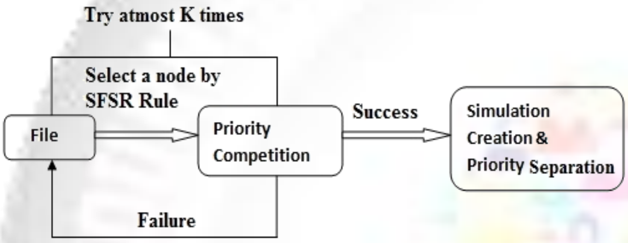
It is a challenge to keep the overall P proportional to the overall $\sum V_{fl}$ while the duplicate files are scattered. We resolve this problem by keeping the average V of the duplicate nodes of a file close to A . Then, Formula (22) can be expressed as

$$\mathbf{n_f} * \mathbf{A} \infty \sqrt{\frac{\mathbf{q_f}}{\mathbf{S_f}}} \Rightarrow \mathbf{n_f} \infty \sqrt{\frac{\mathbf{q_f}}{\mathbf{S_f}}} \Rightarrow \mathbf{n_f} \infty \mathbf{P_f} \qquad (31)$$

In this case SFSR is satisfied. To attain this goal, we let each node deliberately select a neighbor node to create duplicates of its file so that the average meeting ability of duplicate nodes of the file is equal or closest to A . Each node only needs to consider the P of each file in the resource competition. Upon winning a competition for a file, a node separates the file's P evenly between the file and the duplicate. After this, the popularity of each file/duplicate is continuously updated based on the number of requests received for it in a unit time period, which is used to update its priority value P. As a result, the number of simulations of each file is proportional to the sum of contacting ability of its simulation nodes, realizing Formula (22).

B. Design of the File Simulation Protocol

Based on the above solutions, the Priority Based and Separation file simulation protocol (PBS) is proposed.



In PBS, each node dynamically upgrades its meeting ability (V_n) and the nodes mean meeting ability in the system (V). Such details are interchanged among adjacent nodes. Each node also routinely computes the $P_f = \sqrt{\frac{q_f}{s_f}}$ of each of its files.

The process of the duplication of a file in PBS is introduced. Based on SFSR, A node allots priority to its files. Then each node arranges all of its files in descending order of their P_f and creates duplicates for the files in a top-down manner routinely. The pseudo-code for the process of PBS between two contacting nodes is presented by algorithm 1.

When the communication period of the engaged two nodes ends, the duplication for a file stops. Then, the node will continue the duplication process for the file again after removing the disengaged node from the adjacent node list. Since as time goes on, the files popularity and Ps and unoccupied system resource change, and to dynamically manage these time varying features, each node routinely implements PBS.

C. Collecting Information on Contacting Competency

To discover neighbor nodes in MANET, beacon messages are exchanged periodically between nodes, its mobility rely upon the frequency of beacon messages and the size is usually several bytes. To minimize the communication charge, the values V_n and A are piggybacked into beacon messages. In normal MANETs, a node's contacting ability(V_n) is simply determined by the frequency it contacts other nodes. In disconnected MANETs, a node needs to know the distribution of different groups to calculate its performing ability, then each node piggyback its group ID and group information it knows in the beacon message. As nodes contact more and more nodes, the calculated value can generally represent that of all nodes.

Algorithm 1 Pseudo-code of PBS between node n and l.

```
n.createReplicasOn(l)              //node n tries to create a replica on node l
l.createReplicasOn(n)              //node l tries to create a replica on node n
Procedure createReplicasOn(node)
        nCount ←0                                          //initialize a count
        this.orderFilesByP()                    //order files by priority value
        For (each file fi in current node)          //try to replica each file
                If(node.compete4File(fi) == true)            //competition
                        node.createAReplica4(fi) //create a replica if win
                else
                        nCount ←nCount+1
                If(nCount ≥ L )                          //try at most L times
        Break
End Procedure
Procedure compete4File()                              //Compete for file f
        While (nRemainMem < f.size())
                nSum←nTotal←nRandom←fFile ←0          //initialization
                For (each file fi (including f) in current node)
                        nTotal←nTotal+1/P_f
                nRandom← generateARandomNumber() % nTotal
                For (each file fi (including f) in current node)
                        nSum← nSum+1/P_f
                        If(nSum >= nRandom)
                                fFile = fi Break                  //pick the file
                If(fFile = f)          //f is the picked file, competition fails
                        return false
                Else                                     //win the competition
                        select fFile
        delSelectedFiles()                          //delete the selected files
        return true
End Procedure
```

D. Scrutiny of PBS Performance

When a simulation is created for a file with Priority P, then the two copies will simulate the file with Priority P/2 in the next round. After each round the priority value of each file is updated based on the request for the file. During the continuation of process some simulations may be deleted but the total number of request for the file remains the same. Therefore in each round of simulation distribution to nodes based on our design of PBS, the overall probability of creating simulations for original file f, denoted by P_{sf} is proportional to its overallP_f. (i.e.)

$$\mathbf{P_{sf}} \infty \mathbf{P_f} \qquad (32)$$

If suppose, total of M rounds of competition are conducted, the expected number of simulations denoted by n_f for file f is,

$$\mathbf{n_f} = \mathbf{MP_{sf}} \Rightarrow \mathbf{n_f} \infty \mathbf{P_f} \qquad (33)$$

Therefore a conclusion is made that in PBS the number of simulations of each file is proportional to its P, thereby realizing the SFSR.

V. Conclusion

In this paper we have examined the problem of allotting limited resources for file simulation for the purpose of overall superlative file searching efficiency in MANETs. First theoretical analysis is made on the influence of simulation distribution on the average querying detain under limited available resource under two mobility models, and derived an superlative simulation rule to allot the limited resource to file simulation in order to minimize the average querying detain. Existing protocols only consider storage space as resource; we also consider file possessor's ability to contact nodes as available resource since it also affects the average querying detain. This new concept enriches the deduced rule and the effectiveness of the developed simulation protocol. Finally, we designed the Priority Based and Separation file simulation protocol (PBS) that realizes the proposed superlative simulation rule in a fully distributed manner.

REFERENCES

[1] "Qik," http://qik.com/.
[2] "Flixwagon," http://www.flixwagon.com/.
[3] Y. Tseng, S. Ni, and E. Shih, "Adaptive approaches to relieving broadcast storms in a wireless multihop mobile ad hoc network," in Proc. of ICDCS, 2001, pp. 481–488.
[4] B. Chiara, C. Marco, and et al., "Hibop: A history based routing protocol for opportunistic networks," in Proc. of WoWMoM, 2007.
[5] T. Hara and S. K. Madria, "Data replication for improving data accessibility in ad hoc networks," TMC, vol. 5, no. 11, pp. 1515–1532, 2006
[6] W. Gao, G. Cao, A. Iyengar, and M. Srivatsa, "Supporting cooperative caching in disruption tolerant networks." in Proc. of ICDCS, 2011.
[7] J. Broch, D. A. Maltz, D. B. Johnson, Y. Hu, and J. G. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in Proc. of MOBICOM, 1998, pp. 85–97.
[8] M. Musolesi and C. Mascolo, "Designing mobility models based on social network theory," MCCR, vol. 11, pp. 59–70, 2007