This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2007 proceedings.

A Novel Path Selection and Recovery Mechanism for MANETs P2P File Sharing Applications

Ahmed Abada
Dept. of Systems and Computer Engineering
Carleton University
Ottawa, ON, Canada

Changcheng Huang
Dept. of Systems and Computer Engineering
Carleton University
Ottawa, ON, Canada

Li Cui
Institute of Computing Technology
Chinese Academy of Sciences
Beijing, China

Hsiao-Hwa Chen
Institute of Communications Engineering
National Sun Yat-Sen University
Taiwan

Abstract-**Peer-to-Peer (P2P) file sharing protocols for Mobile Ad Hoc networks (MANETS) is becoming an important area of research nowadays because of the constantly escalating demands from consumers to go mobile and the need to share contents between users. Current P2P file sharing protocols for MANETs use either aggressive or reactive flooding techniques for their path selection and recovery. Both techniques represent extreme measures to deal with path failures and they either do too much or very little work in response to path failures. In this paper we present a new path selection and recovery mechanism (Reactive with Unicast Probing Messages) that is a balance between the above two techniques in terms of its control overhead and the resulting path quality. By introducing a new unicast messaging mechanism to the reactive flooding technique, a requesting node can probe its set of available paths for their roundtrip times and use the best one for its file transfer. Our new technique is based on unicast messaging which proves to be effective and requires low control overhead as per the results of our simulations.**

I. INTRODUCTION

Running P2P file-sharing applications [1][2] is generally a challenging task because of the dynamic nature of such applications. Because of the inherited limitations of MANETs, P2P file sharing in MANETs is much more challenging than in conventional wired networks[3][4][5][6][7]. While wired networks only have to deal with a dynamic topology at the application level, MANET networks have to deal with a dynamic topology at both of the application level and the physical level as well as the known wireless limitations (bandwidth, memory, unreliable physical channel, battery and processing power). Because of that, special P2P file sharing protocols like ORION [8] and MPP [9][10] were specially designed to address the MANET requirements.

File sharing applications usually run in two stages, they are the file search and the file transfer. During the file search stage, a requesting node (a node that is interested in downloading a certain file) would flood the network with a QUERY message that have a description of the requested file. Source nodes (nodes that have the requested file) receiving the QUERY message would usually reply back to the requesting node with a REPLY message indicating that they are willing to upload the requested file to it and providing the path that the requesting node should use to reach the source node. Typically, a single QUERY would result in multiple REPLY messages depending on how many source nodes there are in the network. We refer to the way by which a requesting node deals with the set of REPLY messages acquired by the file search stage as the "Path Selection and Recovery" mechanism. Because of the constantly changing topology of the MANETs P2P systems, path selection and recovery mechanisms are used by the requesting nodes to decide which path they are going to use first to download a certain file and how to recover from path failures when they happen. Until now, existing P2P file sharing protocols for MANETs (ORION and MPP) have followed only two techniques for path selection and recovery, they can be referred to as the "aggressive" and the "reactive" techniques. We are going to discuss these two techniques next in detail.

The organization of this paper is as the follows: in section 2 we introduce the existing path selection and recovery mechanisms (aggressive and reactive) used by current P2P file sharing applications, in section 3 we introduce our new path selection and recovery mechanism (reactive with unicast probing messages), in section 4 we present our simulation setup and results, and in section 5 we present our conclusion and future work.

II. RELATED WORKS

The existing path selection and recovery mechanisms can be classified into two general categories: aggressive vs. reactive.

A . Aggressive Path Selection and Recovery

In the aggressive technique (used by the MPP protocol in its Dynamic downloading mode), only the first path acquired by the file search stage is used for file transfer, all subsequently arriving paths are discarded. In the case of a path failure, a new file search (network-wide re-flooding) must be done to get a new path. This technique follows the assumption that the first path acquired by the file search stage will almost always have the best connection to the requested file. Although that this technique always uses the best path available in the network to retrieve the requested file, it usually requires frequent network-wide flooding because a new file search is required each time the used path fails. Existing research [11] have shown that flooding messages on today's wireless networks (using 802.11 standard MAC layer) have a halting effect on the network because nodes overhearing a transmission attempt must refrain from sending any messages until the channel is clear. This waiting process could potentially take a long time if the transmission is taking place over slow physical links making the halting effect on neighboring nodes even greater.

B. Reactive Path Selection and Recovery

In the reactive technique (used by ORION and MPP in its static downloading mode), paths acquired by the file search stage are cached in at the requesting node and are ordered on a first-come-first-serve basis according to the arrival time of their corresponding REPLY messages. They are then used one by one on a sequential basis starting with the first path to be acquired. A network-wide re-flooding to get fresh paths is only required after all cached paths are broken. While the reactive technique uses less frequent network-wide flooding because usually there are more than one path available to be used for file transfer, the requesting node will usually have to deal with slow paths and wait for them to break before it can move one step down the list of available paths and try the next one. This sequential behavior of path selection is there because the requesting node has no way of ordering the set of available paths other than by means of the arrival time of their corresponding reply messages. The main problem with this sequential behavior of the reactive technique is that due to node mobility, the quality of the acquired paths changes over time, and by the time a certain path is to be used, it is usually broken or has a weak connection.

III. PROPOSED MECHANISM

P2P File sharing applications are usually used to download large files, they usually run as background processes and do not need any user input once they are started, so users just need to initiate the downloading process and wait for the P2P application to finish its work. Because P2P applications are meant to be independent most of the time, it is important for such applications to make intelligent decisions about which paths to use for their file transfer especially in the limited resources environment of MANETs.

The main problem with the reactive recovery mechanism is that because of the random node mobility in MANETs, the list of paths available to the requesting node at the beginning of the file transfer stage will not be kept in order over the course of the file transfer. Due to the fact that MANET nodes are always on the go, it is always the case that the quality of the paths gained by the file search stage will fluctuate over time until they eventually break down as a result of node mobility. To achieve better path selection decisions while still avoiding the frequent network-wide flooding associated with the aggressive technique, we have added a new functionality to the reactive path selection and recovery mechanism to rearrange on a periodic basis the set of available paths according to their increasing roundtrip time. Rearranging the set of available paths allows for intelligent switching between file transfer paths so that the one with the least roundtrip time is always used. As a result, no unnecessary waiting for a weak path to break is required if a better one is available. Since only those paths already known (acquired by the file search stage) to the requesting node are being reordered periodically, unicast probing messages are used to keep the needed control messages at a minimum. Because of the limited nature of MPP since it only works with the DSR network layer protocol, we have decided to work with the reactive path selection and recovery mechanism in the ORION platform to provide a more general solution that can work with any MANET network layer protocol. We used the name ORION+ for our enhanced version of the ORION protocol that uses our new path selection and recovery mechanism.

Protocol Details

The ORION protocol employs a file routing table at each node taking part in transferring a file from a source node to a requesting node [1]. File routing tables are used to store all available paths (next hops) acquired by the file search stage and over which, a certain file can be downloaded. Our solution is based on periodically propagating a unicast LINK_QUALITY_REQ message from the requesting node of a certain file to all source nodes of that file (using all available paths for that file in its file routing table). Upon receiving the LINK_QUALITY_REQ message by intermediate nodes that do not have the requested file, they will have to forward the received message to all next hops over which they can reach the source nodes of the requested file (as given by their own file routing tables). Only when a source node of the requested file receives a LINK_QUALITY_REQ for that file then it must send back a unicast LINK_QUALITY_REPLY message along the reverse path towards the requesting node that initiated the LINK_QUALITY_REQ message. Intermediate nodes along all the paths between the requesting node and all possible suppliers of the requested file will evaluate the received LINK_QUALITY_REPLY messages and order the next-hop entries in their file routing tables accordingly.

Since our solution to reorder the set of available paths is based on monitoring the roundtrip time of each path, the file routing tables are modified to include two new time values (TS1 (Time Stamp 1) and TS2 (Time Stamp 2)) for each available path so that the needed roundtrip time can be calculated. The first time-value (TS1) is set to the current time when a node generating or forwarding a

LINK_QUALITY_REQ message sends out that message to a specific next-hop. The second time-value (TS2) is set to the current time when a node receives a LINK_QUALITY_REPLY from one of the next-hops that it has previously sent a LINK_QUALITY_REQ to. The roundtrip time for each next-hop (TS2-TS1) represents the path quality and is used as the basis for ordering the available paths so the best one can always be used. Figure 1 illustrates the relative position of our new path selection and recovery mechanism used in ORION+ (Reactive with unicast probing messages) with respect to the conventional ones (aggressive and reactive).

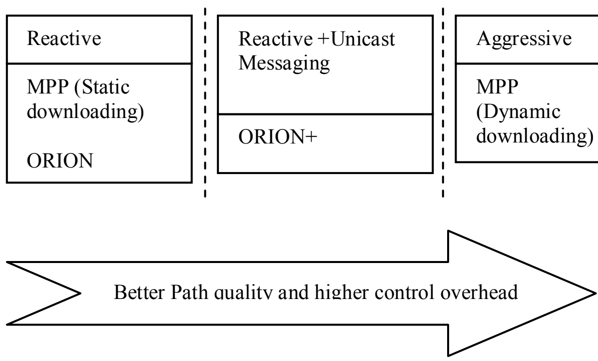

Figure 1. Tradeoff between different path selection and recovery mechanisms

IV. SIMULAION RESULTS

To measure the performance improvement introduced by our path selection and recovery mechanism (represented by ORION+) against the plain reactive path selection and recovery mechanism (represented by ORION), we have used the NS-2 simulator to do some simulation experiments similar to those that were done on ORION. We have performed all our simulations using the ns-2 version 2.29 with the ad hoc network extension from Carnegie Melon University monarch project. We have kept most of our simulation parameters in-line with those used to measure the performance of the ORION protocol so we can compare our results against those published for ORION.

A . Simulation Model

In order to get meaningful results from our simulation, we needed to make the link quality between wireless nodes to vary depending on the physical distance between them. Unfortunately the standard ns-2 simulator does not provide this functionality. In the standard ns-2 implementation, each node has a transmission range and nodes can only communicate with each other if they are within the transmission range of each other. Moreover, the link quality between mobile nodes does not depend on the distance between them, the link either exist if the nodes are within the transmission range of each other or does not exist if they are not within each other's transmission range. To overcome this problem, we have used the OAR multi-rate MAC layer extension provided by the RICE Networks Group (RNG) of RICE University. Combining the Multi-Rate MAC with the transmission range provided by ns-2, now the quality of the link between any two mobile nodes will depend on the physical distance between them until it goes away completely when the nodes go out of the transmission range of each other. The used simulation parameters are as shown in Table 1.

TABLE I
SIMULATION PARAMETERS

Parameter	Set Value
Node transmission range	115 m
Number of nodes	50
Number of files	10
Frequency of LINK_QUALITY_REQ	5 s
File sizes	3 MB
Simulation area	1000 m x 1000 m
Mobility model	Random waypoint
Maximum speed	4 m/s
Rest (Pause) time	5 s
MAC protocol	IEEE 802.11 (with OAR Multi-Rate extension)
Radio propagation	Two-way ground
Network layer protocol	AODV

B. Simulation Setup

In our simulation we used 50 nodes network occupying a 1000 m x 1000 m area. The mobility of the nodes is according to the random waypoint mobility model. All simulation parameters are set according to Table 1. Each scenario was repeated 100 times with randomly generated node movement scenarios. We used 10 sharable files of size 3 MB which is an average size of MP3 songs. Each sharable file is replicated two times at the beginning of each simulation run and nodes are required to download all the 10 files from each other. We have calculated 95% confidence intervals and indicated them on each performance curve. In the following section we present our simulation results regarding to the following three metrics: file transfer time, percentage of successful files transferred, transmitted data volume.

Transfer Time

Transfer time refers to the duration of time it takes from the beginning of the file transfer phase until the whole file is successfully downloaded. From the results in Figure 2, we can see that the original ORION protocol incurs larger transfer times than ORION+ due to the fact that it is reactive in nature. For low mobility, the transfer time goes up for both of ORION and ORION+ because when all remaining available paths are weakly connected, they take longer time to completely break and during that time they provide slower connections. Despite that, ORION+ is able to achieve lower transfer times compared to ORION because of its better path selection mechanism while better quality paths exist in the network. When increasing the speed of the mobile nodes and forcing them to run out of next hops in their file routing tables and hence generating file search phases more often, the entries at the file routing tables become more relevant causing the transfer time to go down but of course global searches must be done more often.

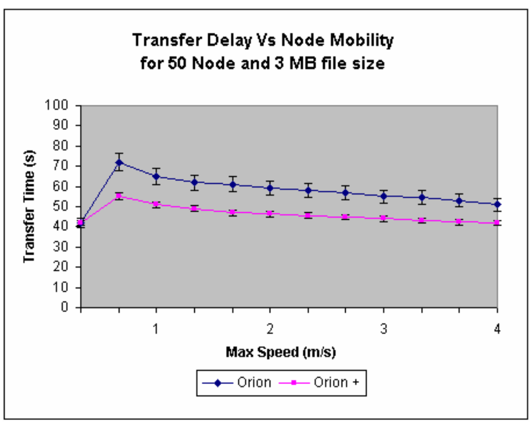

Figure 2. Transfer Delay vs. Node Mobility
(Results are displayed with their 95% Confidence Intervals)

Successful File Transfer

Successful file transfer is a very important attribute from the end-user's point of view because unless the whole file is downloaded successfully, it can't be useful to the user. As it was defined in [1], the download for ORION is considered "failed" if the requesting node runs out of alternative paths after a single re-query. From Figure 3, we can see that ORION+ is able to complete the download process of more files before it runs out of next hops. This is mainly attributed to the better choice of paths while they exist instead of using a specific path until it breaks. Under no mobility conditions, the success rate for both of ORION and ORION+ are the same due to the same connectivity condition for both.

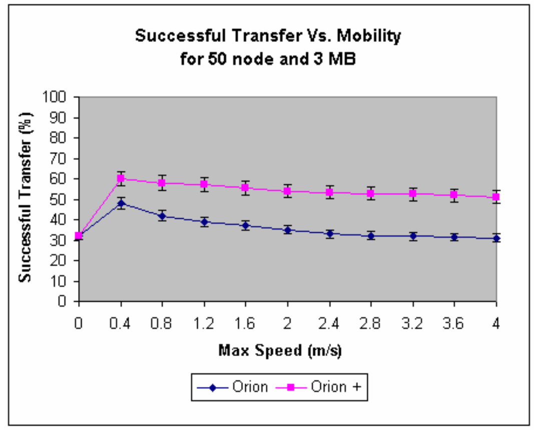

Figure 3. Successful Transfer vs. Node Mobility
(Results are displayed with their 95% Confidence Intervals)

Transmitted Data Volume

Due to the scarcity of available bandwidth in MANETs, it is important that any new feature we add to the ORION protocol comes at a reasonable price. It is hence very desirable to maintain our control traffic at a minimum so that the majority of the available network bandwidth can be used to transfer payload traffic. In this section we compare the total generated traffic injected in the network when using ORION and ORION+. We compare the total transmitted data volume by measuring how many messages were generated in the network for each file transmitted from a source to a destination under each protocol. To do that, for each file transmitted from a source to a destination, we calculate how many data and control messages were generated in total for that file times the number of intermediate hops they went through. For example, if a certain message had to travel three hops to reach its destination, the total number of messages generated in the network is thrice as the original message.

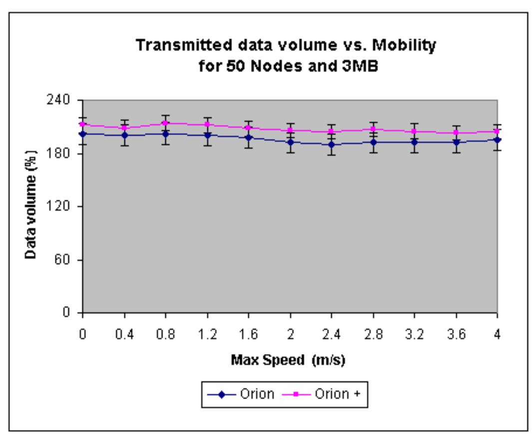

Figure 4. Data volume vs. Node Mobility
(Results are displayed with their 95% Confidence Intervals)

With the actual file size being the absolute minimum for the transmitted data volume (assuming that the file is transmitted directly between the source and destination with no control overhead), we plotted our results to show the total transmitted data volume as a percentage of the actual file size. From Figure 4, we can see that ORION+ does not generate considerable traffic overhead more than the original ORION because its unicast messages are transmitted to a selected number of nodes and does not use flooding.

V. CONCLUSION

P2P file sharing applications are a very convenient way to provide content based routing in MANET environments where no central lookup server is available. However, many considerations have to be taken into account when designing P2P file sharing protocols for the MANET environment, mainly because of the inherited limitations of MANETs and the dynamic nature of P2P file sharing applications. Since path failures usually happen in MANETs P2P systems as a result of the dynamic physical topology, current P2P applications for MANETs have mainly used two path selection and recovery mechanisms (reactive and aggressive) to recover from path failures. Both of the reactive and the aggressive path selection and recovery mechanisms represent extreme measures to deal with path failures in MANETs P2P systems. While the aggressive mechanism seeks to always use the best path available in the network for file transfer, it usually has to initiate frequent network-wide flooding to search for a new path after each failure. On the other hand, the reactive mechanism requires less network-wide flooding because it usually has multiple alternative routes available for file transfer, but the down side of it is that it would occasionally run into weakly connected paths and it does not make any effort to switch to a better connected path even if one existed already in its set of available paths.

In this paper, we presented a new path selection and recovery mechanism for MANETs P2P applications. It was based on the reactive path selection and recovery mechanism in order to avoid the frequent flooding of the aggressive mechanism. We called our new path selection and recovery mechanism “Reactive with unicast probing messages” and used it as the main building block of our new extension of the ORION protocol that we called ORION+. Our new path selection and recovery mechanism uses unicast messages to measure the roundtrip time of each available path so the one with the smallest roundtrip time can be used for the actual file transfer. Our path selection and recovery mechanism have helped to achieve lower file transfer times, higher percentage of successful file transfers and due to the unicast nature of its added control messages, it required minimal messaging overhead as per our simulation results.

Future Work

The new path selection and recovery mechanism (reactive with unicast probing messages) of ORION+ is able to select the best available path for file transfer based on the roundtrip time metric. But when all remaining paths are weakly connected, the system will have to wait for all of them to break (during that time it will be using the best available path) before it can initiate a new file search to get another set of fresh paths. A good direction for future research would be to establish a threshold level of acceptable roundtrip time, so that a new file search (flooding) would be required if no remaining paths are able to provide a roundtrip time lower than that of the threshold level. By doing so, we will further avoid using paths with longer roundtrip time and only initiate file search flooding when needed based on the acceptable threshold. Of course the question that needs to be solved here is how to calculate such a threshold in the totally random environment of MANETs.

REFERENCES

[1] S. Androutsellis-Theotokis and D. Spinellis, “A survey of peer-to-peer content distribution technologies”, *ACM Computing Surveys*, 36(4): 335-371, December 2004.

[2] C.-H. Ding, S. Nutanong, and R. Buyya, “Peer-to-Peer Networks for Content Sharing”, *Peer-to-Peer Computing: Evolution of a Disruptive Technology*, R. Subramanian and B. Goodman (editors), ISBN: 1-59140-429-0, 28-65pp, Idea Group Publisher, Hershey, PA, USA, 2005

[3] Y. C. Hu, S. M. Das, and H. Pucha, “Peer-to-Peer Overlay Abstractions in MANETs”, In *Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless and Peer-to-Peer Networks*, edited by J. Wu, CRC Press, 2004.

[4] A. Hayes and D. Wilson, “Peer-to-Peer Information Sharing in a Mobile Ad Hoc Environment”, in *Proceedings of Sixth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'04)*, 2004.

[5] R. Schollmeier, I. Gruber and M. Finkenzeller, “Routing in Mobile Ad Hoc and Peer-to-Peer Networks. A Comparison”, *In Networking 2002, International Workshop on Peer-to-Peer Computing*, Pisa, Italy, 19.-24.05.2002.

[6] G. Kortuem, “Proem: A Middleware Platform for Mobile Peer-to-Peer Computing”, ACM SIGMOBILE Mobile Computing and Communications Review (MC2R), Volume 6, Issue 4 (October 2002), Pages: 62 - 64.

[7] J. Eberspächer, R. Schollmeier, S. Zöls, and G. Kunzmann, ”Structured P2P Networks in Mobile and Fixed Environments”, *In proceedings of HET-NETs '04 International Working Conference on Performance Modelling and Evaluation of Heterogeneous Networks*, July 2004, Craiglands Hotel, Ilkley, West Yorkshire, U.K.

[8] A. Klemm, C. Lindemann, and O. Waldhorst, “A Special-Purpose Peer-to-Peer File Sharing System for Mobile Ad

Hoc Networks", *Proc. IEEE Semiannual Vehicular Technology Conference (VTC 2003-Fall)*, Orlando, FL, October 2003.

[9] I. Gruber, R. Schollmeier and W. Kellerer, "Performance Evaluation of the Mobile Peer-to-Peer Protocol", *In Proceedings of Fourth International Workshop on Global and Peer-to-Peer Computing*, April 19 - 22, 2004, Chicago, USA.

[10] R. Schollmeier, I. Gruber, and F. Niethammer, "Protocol for Peer-to-Peer Networking in Mobile Environments", *In Proc. of 12th IEEE International Conference on Computer Communications and Networks,* Dallas, Texas, USA, October 2003.

[11] B. Awerbuch, D. Holmer, and H. Rubens, "Effects of Multi-rate in Ad Hoc Wireless Networks", Technical Report, www.cnds.jhu.edu/archipelago/.