This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2009 proceedings.

A Multicast Approach for Peer-to-Peer Content Distribution in Mobile Ad Hoc Networks

Sidney Santos Doria
Marco Aurélio Spohn
Federal University of Campina Grande
Computing Systems Department
Campina Grande, PB - Brazil
{sidney, maspohn}@dsc.ufcg.edu.br

***Abstract*—There are synergies between Mobile Ad Hoc Networks (MANETs) and Peer-to-Peer (P2P) networks. Some previous works consisted on the straight implementation of standard P2P content distribution protocols over MANETs. However, we ponder that such a direct approach may not be the best choice. We propose a novel P2P content distribution protocol for MANETs: Peer-to-MANET (P2MAN). P2MAN takes into account the main characteristics and constraints of MANETs, while keeping useful concepts from P2P networks. P2MAN leverages on the PUMA multicast routing protocol, delivering reliable content distribution at the application layer. PUMA was chosen as P2MAN's routing protocol due to its best performance compared to other representative multicast routing protocols. P2MAN was implemented and evaluated in the NS-2 network simulator, and its source code is publicly available. Extensive simulation results in typical MANET scenarios show that P2MAN is efficient and scalable. P2MAN good performance is mainly due to its multicast approach, applying multicast meshes for content distribution, and a basic repair mechanism for reliably sharing content among peers.**

I. INTRODUCTION

Peer-to-peer (P2P) networks and mobile *ad hoc* networks (MANETs) have a common challenge: design collaborative efforts to provide communication in dynamic and decentralized environments. A combination of MANETs' easy deployment and the resilience of P2P networks has driven the research on the synergy between these networks. The first attempts in that direction consisted on straight implementations of representative P2P protocols over MANETs [1]–[5], analyzing how P2P content distribution protocols, originally designed for wired networks, perform over wireless *ad hoc* networks. As expected, the performance results show that such protocols perform poorly, mainly due to the dynamic characteristics (e.g., topology changes) of MANETs.

Most P2P networks are designed for the Internet (i.e., global scale, with millions of nodes), while typical MANETs are composed of tens to hundreds of nodes. P2P protocols usually work in the application layer, applying unicast for data dissemination, and with no concerns to node mobility. On the other hand, MANETs usually make use of unreliable broadcast transmissions over a shared channel, while nodes are let to move about freely. These contrasts have challenged the research on optimizing P2P networks over MANETs [3], [6], and some proposals [3], [6]–[8] follow a cross-layer approach between the network layer and the application layer, breaking the standard OSI encapsulation model.

We ponder that a perfect symbiosis between P2P and MANETs requires more than a direct implementation with a few adaptations. The choice for a cross-layer approach is already an indication that this is a complex endeavor. To show that it is possible to take advantage of P2P concepts in MANETs, we present a multicast based approach, *Peer-to-MANET* (P2MAN), for providing content distribution in MANETs.

P2MAN leverages on the PUMA [9] multicast routing protocol, delivering reliable content distribution at the application layer. PUMA is a mesh based receiver initiated multicast routing protocol. PUMA was chosen as P2MAN's routing protocol due to its best performance compared to other representative multicast routing protocols.

P2MAN is inspired on modern P2P content distribution protocols, but it takes into account the special characteristics and constraints of MANETs. Section II describes P2MAN in detail. The novelty in P2MAN derives from its use of a very simple and low control overhead multicast protocol for the content distribution, and by applying a single multicast channel for accomplishing all the control functions. An extensive simulation of P2MAN using the *Network Simulator 2* (NS-2) [10] is presented in Section III. We perform experiments addressing the impact of concurrent requesting nodes, node mobility, and the content size. The results point out that P2MAN provides scalable and efficient content distribution in MANETs. In addition to that, P2MAN source code is publicly available for those who want to try and extend it. As an additional contribution, PUMA [9] as our choice for the multicast protocol was also implemented in the NS-2, and we have made the PUMA source code publicly available. Section IV concludes this work.

II. PEER-TO-MANET

Peer-to-MANET (P2MAN) is a multicast based content distribution protocol for mobile *ad hoc* networks. P2MAN is designed to mitigate MANET's constraints, achieving good performance by taking advantage of P2P concepts and MANETs' capabilities. P2MAN uses multicast groups to deliver contents, and a special multicast group, called *Public Channel*, to accomplish all control functions. In this paper, we restrict contents to digital objects (i.e., files), like those shared in popular P2P content distribution networks.

When a P2MAN node starts, it joins the *Public Channel* (PC) for exchanging control messages. All peers willing to share any content are PC members. Content location is not cached at any node. Instead, to search for any content, a node queries the *Public Channel* without the need for a network-wide flooding of control messages. In case any active peer has the content, the peer is reachable through the PC group, and a reply is propagated back through the public group. The reply also carries metadata generated by the content owner, having detailed information about the object (i.e., file).

Like in most P2P networks, P2MAN slices content for delivery. The owner decides how the object will be splitted in pieces, representing an object image using a bitmap (where each bit represents an object slice). The owner also decides which multicast group will be used for transmitting the object. The metadata is created by the owner, containing the necessary information for guiding the requesting nodes. After receiving the reply, the requesting node joins the respective multicast group and sends an authorization message to the PC group. By receiving the authorization message, the content owner can initiate the object transmission. Considering that we assume an unreliable routing musticast protocol (with no transport layer multicast protocol!), reliability is granted at the application layer by employing a retransmission mechanism provided by the *Repair Mode*. On Repair Mode, a receiver node can claim for pieces not received yet. The owner then retries the transmission of the remaining pieces. While there are missing pieces, new claims and retransmissions will take place until the receiver gets all the pieces. In this work, security issues are not addressed at any layer.

A. *Content Search*

The PC group is a special multicast group for exchanging control messages. A message sent to the PC group means a scoped flooding restricted to the P2MAN nodes. The PC group address is pre-defined, and we assume all nodes know it. A peer which owns a complete copy of a shared object is referred to as the *owner node*. A peer requesting any object is referred to as the *requesting node*.

A requesting node sends a *Request Message* (RM) to the PC group. An RM packet carries the content ID (i.e., the object name), and because it is sent to the PC group the message potentially reaches all active peers. In case the requesting node does not get any reply from other peers after some period of time, the RM message is repeated until a reply is received or the maximum number of retries is reached, after which the node gives up. The number of retries for an RM packet can be tuned appropriately. When an owner node receives the RM packet, the node replies by sending a *Search Reply* (SR) message. An SR packet carries metadata regarding the requested object, and it is sent via the PC group.

An object metadata contains all the necessary information for accomplishing the content distribution. This metadata guides the requesting nodes, describing how the object is going to be splitted, size of each object slice, and how many slices make the object. Similar to the *BitTorrent* protocol [11], P2MAN maps objects' pieces to a bitmap. The owner node also decides which multicast group will be used to transmit the content. Metadata regarding any object includes the following information:

- Content (i.e., object) name;
- Transmission multicast group address;
- Number of pieces;
- Size of each piece;
- Bitmap.

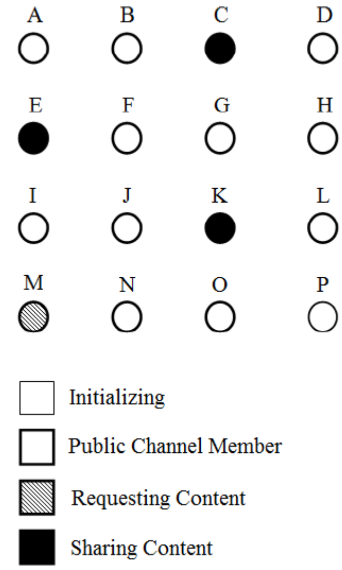

Fig. 1. P2MAN Scenario with Nodes' State.

In case the RM packet reaches multiple owner nodes, all of them will send an SR packet to the PC group. The requesting node may receive all the reply messages. To make things simpler, the requesting node will take into account only the first reply, disregarding any other SR packet sent in response to the same request. We have adopted this approach because the first reply is usually the best one, assuming it was sent from a closer node or through a better path (i.e., in terms of congestion or contention level). But notice that all replies must be sent to assure diversity. In addition to that, there is no guarantee that an SR packet will eventually reach its target; therefore, redundancy is welcome in such a dynamic environment, specially when considering that only owner nodes introduce redundancy, then it is probably within an acceptable level.

Figures 1 and 2 illustrate the content search process. Nodes `E` and `K` both own a copy of an object that they are willing to share. At some point, node `M` desires that content. To get the object, node `M` sends an RM packet to the PC group, querying other P2MAN peers currently active in that group. Nodes `E` and `K` receive the RM packet and reply back with an SR message sent to the PC group. Each SR message carries

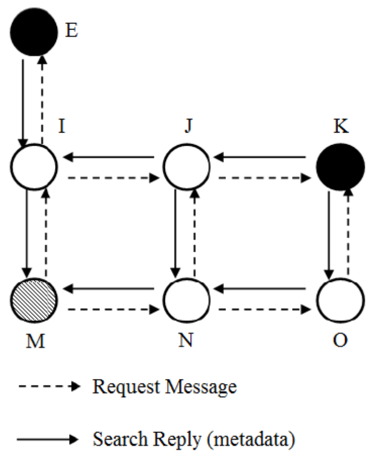

Fig. 2. Searching for an object.

metadata from the respective owner node. The replies are transmitted hop-by-hop to all P2MAN peer nodes, but due to the dynamics of the mobile network the requesting node might receive both replies, one, or none of them. In our example, node `M` receives both replies, but chooses the one received from node `E` considering this one was received first. The late reply from node `K` is just thrown away.

B. Content Delivery

Once a requesting node receives the first SR message, the node checks the metadata, joins the designated multicast group, and sends an *Authorization Message* (AM) to the PC group. An AM packet carries the ID (e.g., the IP or MAC address) of the owner selected by the requesting node. By sending out the AM message to the PC group, it lets any other replying owner release multicast groups reserved for eventually sending the requested object. When the selected owner receives the AM packet, the node starts transmitting the object blocks through the multicast group reserved for that object transmission. Notice that the owner node does not join the target multicast group, because the node is just a sender. Given that it is a stateless protocol, after finishing transmitting the content, the owner just stays silent.

After sending the AM packet, the requesting node starts a timer for receiving the first object fragment. The timer is restarted every time a new piece is received. In case of timeout, the receiving node goes into *Repair Mode* (explained in detail in Subsection II-C). A timeout with missing blocks means that there are transmission problems (e.g., collisions, congestion) going on. Once the requesting node gets all object fragments, the node can itself start sharing the object with other peers, what is recommended to ensure diversity.

As a multicast based protocol, P2MAN is designed to perform well in one-to-many transmissions. This is not unusual in the context of content distribution networks, because some nodes can concurrently be interested in the same content. If that is the case, requesting nodes will send RM packets regarding the same content to the PC group. All reachable owners will reply to each RM packet, sending the appropriate metadata to guide all requesting nodes. However, we have chosen to make the owner sending only to the first requesting node. All the other requesting nodes will receive pieces opportunistically, because they will all get to know about the object transmission through the PC group.

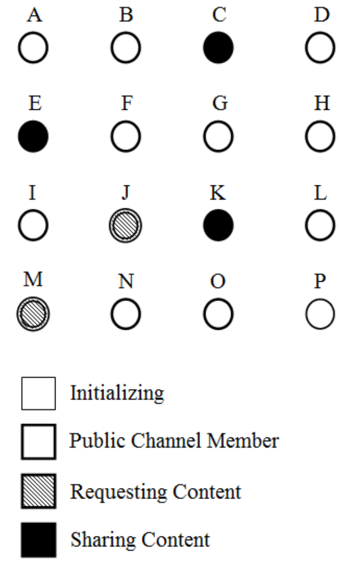

Fig. 3. Ready to Receive.

Figures 3 and 4 illustrate the content delivery process. In Figure 3, nodes `J` and `M` are concurrently requesting the same object. They searched for the content and received replies from nodes `E` and `K`. Assuming that both received the first reply from node `E`, nodes `J` and `M` join the target multicast group announced by node `E`. After that, the requesting nodes send AM packets to the PC group, announcing node `E` as the selected sender. Node `K` also receives the authorizations, but it releases any allocated resources (e.g.,multicast group address) resulting from the queries received from the requesting nodes. In our example, node `E` receives the first AM packet from node `M`. However, given that both nodes `J` and `M` joined the same group announced by the sender, they are going to receive the object' fragments through a single multicast transmission.

If node `J` had chosen node `K` as the source instead, there would be two multicast transmissions simultaneously. Even though it might sound like a waste of resources, in some cases it pays off to have some redundancy given that all packets are broadcast throughout the multicast mesh built by the multicast protocol (i.e., we have chosen to use the PUMA [9] routing protocol given its better overall performance compared to other representative multicast protocols).

In most P2P content distribution networks, multiple nodes transmit simultaneously different pieces of the same object to improve the capacity of the network. In fact, keeping nodes transmitting is the main reason for the P2P incentive mechanisms. However, in P2MAN, we propose the application of unreliable multicast transmissions instead of reliable unicast

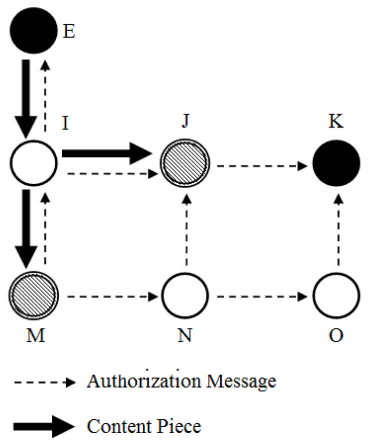

Fig. 4. Receiving Pieces.

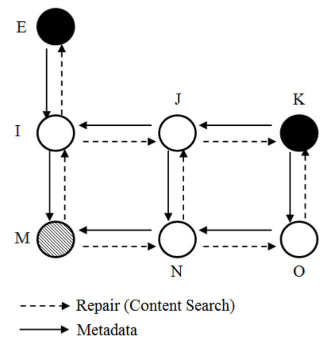

Fig. 5. Single node in repair mode.

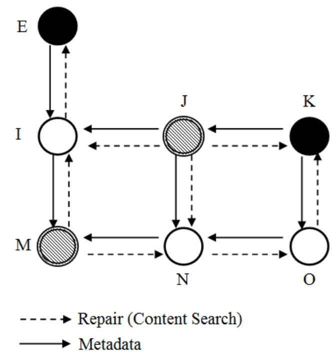

Fig. 6. Multiple nodes in repair mode.

transmissions (usually costly due to the intrinsic characteristics of multihop wireless communications). Likewise, we have not considered a reliable transport protocol because it has already been shown that TCP like protocols [12], [13] do not perform well in multihop wireless networks (i.e., packet losses due to link failures are misunderstood as congestion). On the other hand, diversity is enforced during the owner selection process. Due to the multicast approach, another purpose for the authorization process is also the guarantee that only one node will transmit the content to the multicast group members.

Despite of P2MAN be intended for typical MANET scenarios, multicast addresses might get depleted after several content transmissions. At its current state, P2MAN does not take into account multicast address recycling. However, multiple object transmissions could well share the same multicast group, given that all control messages exchanged through the PC group carry content metadata. It is like the receiving node tuning on the corresponding channel for receiving object pieces being transmitted at the present moment, while the node can always request the missing pieces later on using the repair mode. A node can define multicast addresses based on its unique IP address, avoiding multiple nodes choosing the same multicast address.

C. Repair Mode

Repair Mode is a mechanism to ensure content delivery reliability. Basically, a node switching to Repair Mode restarts the content search process. However, the RM packet will now carry a bitmap showing which pieces the requesting node has already received. Therefore, an owner node will have to send only those missing pieces. In the search process, the requesting node can well choose a source different from the previous one. If this happens, a new multicast group will be used for the transmission, and the requesting node will have to leave the old multicast group and join the new one.

If several nodes are receiving the same content, some of them may eventually need to repair their transmissions, and a sender could receive many RM packets in response to missing object blocks. Similar to an usual content search procedure, the owner node will consider only the first RM packet. Given that all requesting nodes are PC members, they will eventually get to know about the content distribution. After the transmission of the requested content, nodes still needing any missing pieces will request them via repair mode once again. Assuming that the network is connected, requesting nodes will eventually gather all object' pieces after a finite number of repair rounds, which depends on the average number of missing pieces.

As an example, consider that node `A` needs pieces 2 and 3, and that node `B` needs piece 5. In this case, two repairs will be needed. However, if node `B` needed piece 3 instead, only one repair would take place: in response to the repair initiated by node `A`, node `B` will get its missing piece.

Figures 5 and 6 illustrate the repair mechanism, showing two possible scenarios: (a) A single node (i.e., Node `M`) starting the repair process (Figure 5), and (b) nodes `M` and `J` concurrently start repairing their transmissions (Figure 6). In the second scenario, it is possible that both requesting nodes choose different sending nodes. However, assuming that they both choose the same, there will be a single multicast transmission to repair both incomplete content distributions.

D. PUMA Multicast Protocol: a short description

We have chosen PUMA [9] as our multicast routing protocol, because PUMA has been shown to outperform the most representative multicast routing protocols. However, at the time we started this work, there was no PUMA implementation publicly available. As an additional contribution to this work, we have implemented PUMA for the *Network Simulator 2* (NS2), and we have made its source code publicly available [14].

PUMA is a receiver initiated mesh based multicast protocol. By default, the first receiver in a multicast group acts as the core group. In case multiple nodes simultaneously join the same group, the node with the largest IP address is actually elected the core. What makes PUMA simple and very efficient is its low control overhead. A single control message, a multicast announcement, is used to maintain the mesh. Besides that, multiple meshes can be compiled into a single announcement. PUMA does not require any unicast protocol, and all transmissions are broadcast. Even though broadcast

TABLE I
P2MAN: SIMULATION PARAMETERS.

Parameter	Description
Simulator	*NS-2* [10] Release 2.33
Number of rounds (samples)	10
Terrain size	1000m x 1000m
Mobility model	Random Waypoint
Pause time	0 s
Radio range	250 m
Bandwidth	2 Mbps
MAC protocol	802.11 DCF Mode
Content pieces	500 Bytes and 1000 Bytes

transmissions are unreliable, the mesh itself introduces some redundancy, and because the mesh includes only group members and the nodes interconnecting them, we actually have scoped broadcasts within the mesh.

As a multicast announcement propagates throughout the mesh, nodes learn the shortest path to the core. This way, data packets can be quickly routed to the core. On its way toward the core, two things can happen to a data packet: (a) the packet goes all the way until it reaches the core, or (b) a mesh member is hit before reaching the core. Anyway, once a data packet reaches the mesh, the packet propagates only inside the mesh. The core is not a single point of failure, because when the core fails the group member with the largest IP address takes the core role.

III. PERFORMANCE ANALYSIS

Despite all control messages sent via the PC group, only peers take part of such group, and considering that PUMA builds a mesh which includes only the peers and the nodes interconnecting them, we observe a scoped flooding within the PC group only.

We have implemented the P2MAN protocol and the PUMA multicast protocol in the *Network Simulator 2* (NS-2) [10], and we have made their source code publicly available [15]. Extensive simulations were run for typical scenarios, addressing the impact of concurrent requesting nodes, node mobility, and the content size. For each scenario and configuration, results represent the average over ten rounds, considering a 95% confidence interval. Nodes were randomly spread over a 1000m x 1000m terrain, and they move according to the Random Waypoint mobility model (excluding minimum speed of zero). Shared objects are sliced into pieces of 500 Bytes and 1000 Bytes (following recommendations by Lee et al. [16] regarding packet sizes). Table I shows the main simulation parameters.

First of all we analyze how P2MAN performs when varying the number of concurrent requesting nodes. Considering that we are using multicast to deliver content, the average content delivery time should vary smoothly as we increase the total number of group members (i.e., requesting nodes). However,

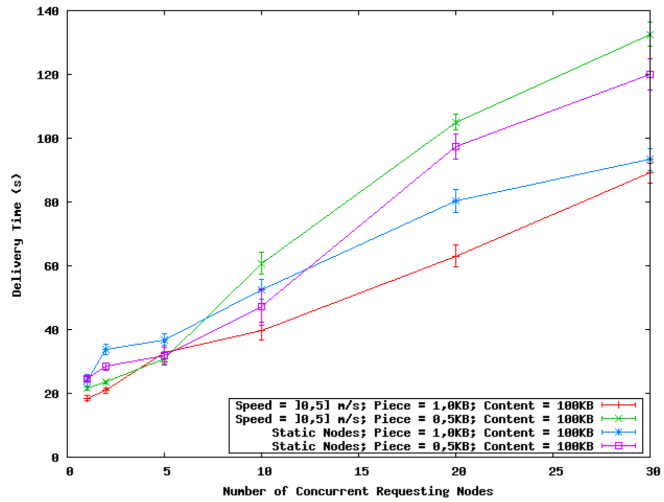

Fig. 7. Delivery Time x Requesting Nodes.

considering that we have many factors affecting the dynamics of the network (e.g., node mobility, and average mesh diameter), the average content delivery time is expected to vary proportionally with the number of requesting nodes.

We considered a network with fifty nodes. Among these nodes, one is randomly selected as the content provider (i.e., owner node). Content size is defined as 100 KB. At some point, some requesting nodes start the content discovery process. The number of concurrent requesting nodes varies from one to thirty. Each simulation is run with mobility (with maximum speed set to 5 m/s) and without mobility. To investigate the impact of content piece size, simulations are run for pieces of 500 Bytes and repeated for pieces of 1000 Bytes.

The results (see Figure 7) show that the average delivery time varies linearly with the number of requesting nodes. P2MAN performs well in all scenarios under consideration. Delivery time was poorly affected by mobility or content piece size. In fact, mobility and larger pieces (i.e., 1000 Bytes) actually improved the overall performance.

Figure 8 shows the simulation results when varying the content size (i.e., shared object size). Once again, we observe that delivery time increases linearly with the content size. Even though there is an increase in the number of transmission repairs, because there are more packets being transmitted (i.e., more contention), it is shown to be more efficient than unicast based repair mechanisms due to the fact that a single repair may well recover multiple broken transmissions. However, repair mode reduces P2MAN overall efficiency, because there are redundant transmissions (i.e., pieces already received by a subset of nodes). For future work, it is possible to explore situations when it is worth unicasting pieces to single failing nodes. There is also a tradeoff related to how long a node should wait before resorting to repair mode: short waiting periods may lead to unnecessary repairs, while waiting for too long might increase average delivery times. In the current implementation, waiting time is set to 400 ms.

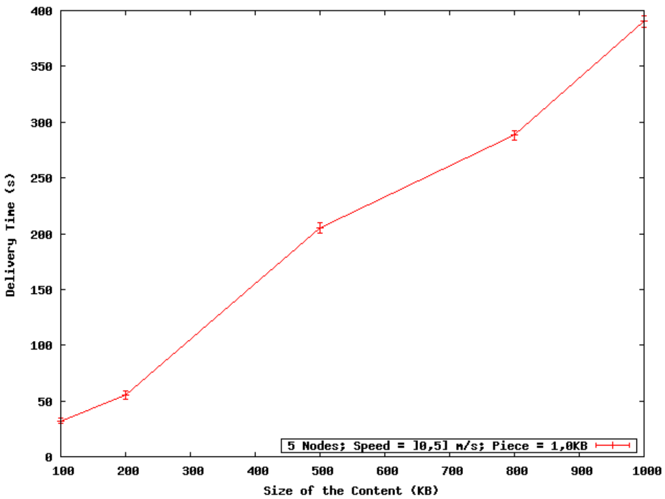

Fig. 8. Delivery Time x Content Size.

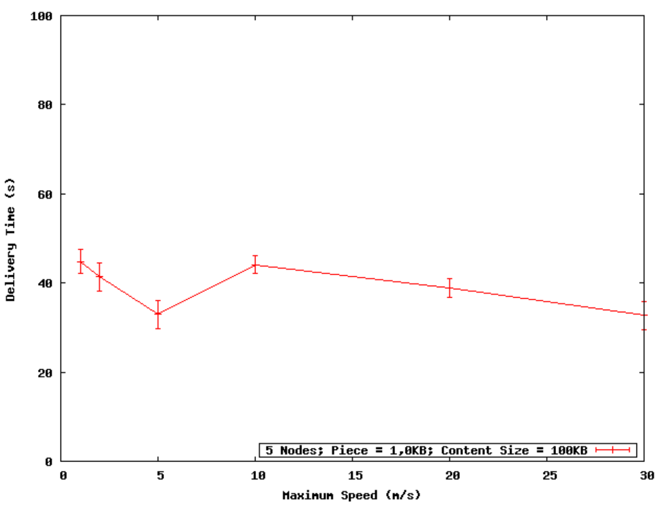

Fig. 9. Delivery Time x Mobility.

Figure 9 shows the simulation results when varying node mobility, assuming object size of 100 KB, and five concurrent requesting nodes. Average node speed was increased from 1 m/s to 30 m/s. The results show that mobility does not impact the protocol efficiency, because transmissions are broadcast within the mesh which is dynamic and adapts quickly due to its low control overhead (i.e., a single control message is necessary to update the mesh!).

IV. Conclusions

In this paper we have presented *Peer-to-MANET* (P2MAN), which is a low overhead multicast based peer-to-peer content distribution protocol. P2MAN leverages on the PUMA multicast protocol, which had already been shown to have a way better performance compared to other representative multicast routing protocols. A single mesh, named *public channel* group, is used for bringing peers together and for discovering shared content. No unicast protocol is needed, and only broadcasting is used for all control and data transmissions. Given that peers organize themselves within multicast meshes, broadcast is scoped to within the mesh, reducing the control overhead, and taking advantage of the natural diversity brought by meshes. Reliable transmission is accomplished at the application layer, through a basic repair mechanism. Based on extensive simulation results, P2MAN is shown to provide good performance results when varying the total number of requesting nodes, shared object sizes, and node mobility. As an additional contribution, we have made all source code, for both P2MAN and PUMA (to the best of our knowledge, this is the first PUMA public implementation), publicly available.

References

[1] G. Kortuem, J. Schneider, D. Preuitt, T. Thompson, S. Fickas, and Z. Segall, “When peer-to-peer comes face-to-face: Collaborative peer-to-peer computing in mobile ad hoc networks,” in *Proceedings of the First International Conference on Peer-to-Peer Computing*, 2001, p. 75.

[2] A. Klemm, C. Lindemann, and O. Waldhorst, “A special-purpose peer-to-peer file sharing system for mobile ad hoc networks,” *IEEE 58th Vehicular Technology Conference*, vol. 4, pp. 2758–2763, 2003.

[3] G. Ding and B. Bhargava, “Peer-to-peer file-sharing over mobile ad hoc networks,” in *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, 2004, p. 104.

[4] M. Gerla, C. Lindemann, and A. Rowstron, “P2P manet’s - new research issues,” in *Perspectives Workshop: Peer-to-Peer Mobile Ad Hoc Networks - New Research Issues*, ser. Dagstuhl Seminar Proceedings, no. 05152, Dagstuhl, Germany, 2005.

[5] L. Oliveira, I. Siqueira, and A. Loureiro, “On the performance of ad hoc routing protocols under a peer-to-peer application,” *Journal on Parallel and Distributed Computing*, vol. 65, no. 11, pp. 1337–1347, 2005.

[6] M. Conti, E. Gregori, and G. Turi, “Towards scalable p2p computing for mobile ad hoc networks,” in *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, 2004, p. 109.

[7] Rajagopalan, Sundaram, Shen, and Chien-Chung, “A cross-layer decentralized bittorrent for mobile ad hoc networks,” in *Third Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services*, 2006, pp. 1–10.

[8] U. Kozat, I. Koutsopoulos, and L. Tassiulas, “A framework for cross-layer design of energy-efficient communication with qos provisioning in multi-hop wireless networks,” in *Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 2, 2004, pp. 1446–1456.

[9] R. Vaishampayan and J. J. Garcia-Luna-Aceves, “Efficient and robust multicast routing in mobile ad hoc networks,” *IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, pp. 304–313, 2004.

[10] “The network simulator,” http://www.isi.edu/nsnam/ns/.

[11] B. Cohen, “Incentives build robustness in bittorrent,” in *Workshop on Economics of Peer-to-Peer Systems*, Berkeley, CA, EUA, May 2003.

[12] G. Holland and N. Vaidya, “Analysis of tcp performance over mobile ad hoc networks,” in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, 1999, pp. 219–230.

[13] T. D. Dyer and R. V. Boppana, “A comparison of tcp performance over three routing protocols for mobile ad hoc networks,” in *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing*, 2001, pp. 56–66.

[14] S. Doria and M. A. Spohn, “Puma multicast routing protocol source code for ns-2,” http://sourceforge.net/projects/puma-adhoc.

[15] ——, “Peer-to-manet source code for ns-2,” http://sourceforge.net/projects/p2man.

[16] J. Lee, G. Kim, and S. Park, “Optimum udp packet sizes in ad hoc networks,” in *Workshop on High Performance Switching and Routing. Merging Optical and IP Technologies*, Seoul, South Korea, 2002, pp. 214–218.