Efficient File Sharing by multicast – P2P Protocol using Network Coding and Rank based Peer selection

Tudor M. Stoenescu
Jet Propulsion Laboratory
California Institute of Technology, CA, USA
Tudor.M.Stoenescu@jpl.nasa.gov

Simon S. Woo
Jet Propulsion Laboratory,
California Institute of Technology,
University of Southern California, CA, USA
Simon.S.Woo@jpl.nasa.gov, simonwoo@usc.edu

***Abstract*— In this work, we consider information dissemination and sharing in a distributed Peer-to-Peer (P2P) highly dynamic communication network. In particular, we explore a network coding technique for transmission and a Rank Based Peer Selection method for network formation. The combined approach has been shown to improve information sharing and delivery to all users when considering the challenges imposed by the space network environments.**

Keywords-component; P2P, Network Coding, Rank Based Peer Selection

I. Introduction

In today's communication age there is a growing interest in finding new ways for effective sharing, distribution, and exchange of information content among multiple mobile device users. Originally, these research topics were mainly studied in the context of social group networks and media sharing environments [1], while more recently, they have been investigated in all aspects of communication environments, including highly dynamic vehicular and military communications networks.

Specifically, in the battle field, fast updates on enemy positions, command and control messages, and monitored data are critical. Further, those information need to be shared and new information has to be disseminated among the highly dynamic network elements. Therefore, distributed network architectures and technologies are being investigated for the development of a self-organizing distributed mobile network system for the military use. Such systems have the potential due to their distributed cooperative nature, of being more reliable, versatile, easier to maintain, and capable of adopting fast changing network environment. A major challenge in successfully developing such systems stems from the decentralized and distributed nature of these architectures. In order to ensure their success and safety, new and more efficient coordination and information sharing methods must be developed.

In Fig. 1, we illustrate an example of such a highly dynamic military network, where a server satellite tries to distribute data to other peers. In this example we assume that some of nodes are equipped with multiple antennas enabling it to concurrently communicate to other nodes within its proximity. Due to the highly distributed and decentralized nature of networks, there is a potentially tremendous burden on the information transfer, data sharing and system coordination. For this reason new efficient communication protocols are needed for supporting such systems.

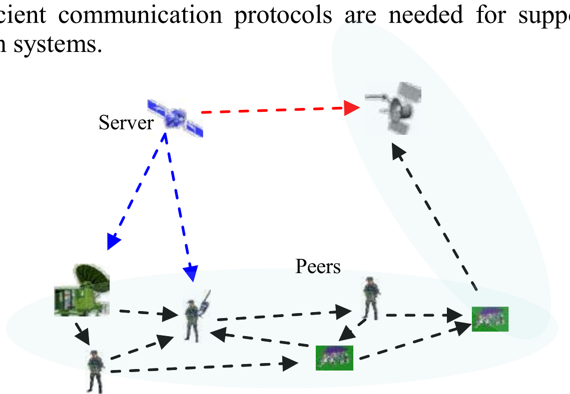

Figure1. Possible military communication networks

To tackle the efficient data distribution over highly dynamic network environments, various Peer-to-Peer communication protocols and strategies have been explored. In particular, due to its vast popularity and success on the terrestrial network, the use of BitTorrent like protocols have been proposed for information dissemination. However, content delivery over wireless communication links brings many new challenges to the problem, such as: a high variability in link conditions due to mobility; interference from multiple conflicting communications; lack of centralized information of the network topology; and the potential need of broadcasting to many users the same content at the same time. In particular, in the case of highly dynamic and challenging military communications with low bandwidth and varied propagation delays, protocols developed for current P2P environments may not be suitable.

In this paper we consider the information propagation problem in a distributed network environment where network structure is dynamic. The main features of the protocol developed in this work are as follows: (i) each user in the network selects its peer based on a file sharing ratio; (ii) data transmitted to each user is the one which is the most scarcely available; (iii) peers are randomly dropped periodically in order to search for new potentially more beneficial peers; and (iv) transmitted data is coded in order to facilitate the concurrent transmission to multiple peers. The main benefits of our approach are:

This research is supported under the title "Improving In Situ Communications using Network Coding" by the Research and Technology Development fund from Jet Propulsion Laboratory.

- Point-to-Point links will only be maintained among users which have a balance of good connection and data discrepancy. This will ensure that the "value" of data transmission will be maximized.
- By communicating scarcely available data we ensure information diversification and an increase of information availability.
- Peers that do not contribute/transmit enough are periodically dropped in order to search/discover new potentially better peers. This will ensure that even with the lack of centralized information, if network environment changes, better new network topologies are discovered.
- By coding the transmission we will be able to communicate various data to multiple users concurrently. This will significantly reduce the number of concurrent transmissions and interference, in turn decreasing the time needed for data propagation in the system.

The coding techniques which will be adopted in this paper will be rather simple XOR codes. The reasons for not considering more complicated codes are three fold: (i) we mainly consider the case in which the network is dynamic and relatively sparse, where each peer can establish the connection with a small number of peers at one time, (ii) full network topology is unknown to each peer and for this reason more sophisticated transmission policies which may require centralized network information may not be possible, and (iii) some of the peers are assumed not to have excess computer power to be able to store and decode complex codes.

The rest of the paper is organized as follows: First we present some of the related work in the literature. In Section III we define some terminology and present our algorithm. In section IV we present some simulation results, and we conclude in Section V.

II. Previous Work

The BitTorrent [1] protocol is one of the most widely used file distribution protocols in the terrestrial network, allowing multiple peers to concurrently download and share files. This protocol, although extremely efficient on terrestrial networks, has not been fully tested for dynamic wireless network environments. In particular the environments on which this protocol has been proven to perform well are ones where the network structure is relatively static, well connected, feedback of message receipt is short and there is at least a minimal amount of centralized information available. In a dynamic network environment such assumptions may not be valid and for this reason this protocol may severely underperform.

Recently, there has been an increasing interest on applying network coding [2] to peer-to-peer communications. Gkantsidis and Rodriguez [3],[4] investigated the benefits of using network coding for large scale content distribution and the advantages of using network coding for various scenarios. These results have been shown through simulation and by real live experiments. Authors in [5-8] exploited the opportunistic XOR network coding techniques in various wireless broadcasting scenarios. Chiu *et al.* [9] derived the theoretical maximum achievable throughput in star P2P networks using network coding. However, only static networks have been analyzed in [9], and they have not considered how the network coding technique may be useful in the dynamic network setting.

The focus in this work is to explore peering strategies along with network coding. We propose a periodic rank based peer selection in order to choose peers so as to maximize the efficiency of bandwidth utilization. We opportunistically apply a simple XOR network coding similar to the ones in [5-8], which gives us the benefit of enabling nodes to perform decoding immediately, without the use of extra buffer for storage. We consider best effort transmission in hop to hop communications since delivering information is critical.

The scheme which we have developed is sender driven, allowing senders to choose its peers in order to maximize content distribution. The proposed scheme is broadly applicable to other wireless communication settings; hence it can be used on highly dynamic vehicular wireless communications links as well as mobile social networks, which incur intermittent connectivity and short contact time.

III. The Algorithm

In this section we are going to present a new P2P algorithm for efficient file distribution in highly dynamic military network. Some of the main features of this algorithm are: it is sender driven, messages are coded in order to maximize information transmission, and peer connections evolve dynamically in order to improve content distribution and discover new more efficient network structures.

A. Definitions and Assumptions

Let $\boldsymbol{S}$ denote the set of users in the network. Denote by $\boldsymbol{F}$ the information/file which the all the users are interested in sharing. Suppose that the file $\boldsymbol{F}$ is broken down into $\mathbf{n} = \{ n_1, n_2, \ldots n_n\}$ chunks of packets. For each user $\boldsymbol{i} \in \boldsymbol{S}$, denote by $\boldsymbol{N_{i,t}} \subset \boldsymbol{S}$ the set of users that $\boldsymbol{i}$ can see at time $\boldsymbol{t}$. This set will be called the set of i's friends at time t.

At any time t, there will be users which may have the complete file available to them (called *seeders*) and users which only have part of the file available to them (called *leechers*).

In this work, we made following assumptions:

1) All users can connect to a subset of users at one particular time. This can be achieved by existing distributed hash algorithms (DHS) in [1]. In this work we do not evaluate specific mobility pattern, but rather we are interested in evaluating the dynamic behavior of networks.
2) At the time of transmission we assume that each sender knows what information each receiver needs. In practice this information can be gathered from the users periodically or through acknowledges of packet receipt.

3) For evaluation purposes we simulate the case when all the nodes are synchronized and communicate packets at the same time. In the case in which this assumption is weakened we do not believe the performance of the algorithm will change much.
4) For simulation purposes we have taken all the links to have the same capacity and transmission delay. In practice, we expect that due to the method in which the algorithm conducts the user's peer selection, the algorithm will adaptively pick the best network topology for information dissemination.
5) For analysis purposes we assume that the number of connections each user may make at one time is fixed to a predefined number d.

B. The Algorithm Description

In this section, we present the transmission algorithm from the perspective of each user. The algorithm works as follows:

1. Initially, user i determines a set of peers to which he/she will connect. This set of peers will be called the set of "friends/peers" of user i and is denoted by $\boldsymbol{N_{i,t}}$.
2. Given that the set of friends has been determined, at each transmission time user i determines a subset d of friends to which to transmit. This set will be denoted by $\boldsymbol{D_{i,t}}$.
3. Given that the set $\boldsymbol{D_{i,t}}$ has been determined (the generation of this set is presented below), user i will code a message in order to transmit useful information to all the d peers. The method of coding the message is presented below.
4. Periodically the users reevaluate the sharing ratio from all of their friends and they drop some of them in order to form new friendships. This policy will enable users to drop the least beneficial users and potentially discover new ones which are more beneficial, and to whom to connect.

C. Choosing the Set of Friends for Transmission

At each time of transmission, user i will pick a set of d friends to whom to try to transmit. This set of friends is represents the set of friends of i to whom i has recently transmitted the least. There are many ways of determining the measure of ranking this amount of transmission. In this paper, the ranking method we are implementing is determined by how much we have transmitted to each individual friend since the last time we have dropped and added new friends. This method seems to be a good one in the case in which the simulation process assumes synchronicity of the decisions and transmissions. In cases in which things tend to be asynchronous, other ranking processes such as the sum of discounted weights based on how long since a transmission was made to each friend may be a better way of determining this ranking system.

D. Coding of the information

For coding of the information to be transmitted we have decided to opt for a simple bit-wise XOR coding technique which does not require heavy computation or additional data storage. This technique works as follows:

1. Upon determining the set of friends to whom to transmit at time t the information, user i determines the packets which are the least available among all of his $\boldsymbol{N_{i,t}}$ friends. If one of this least available packets is unavailable to all $\boldsymbol{D_{i,t}}$ then that packet is transmitted.
2. If a no packet that is least available to the $\boldsymbol{N_{i,t}}$ friends is unavailable to all the $\boldsymbol{D_{i,t}}$ friends, then we look to see if we can find two least available packets such that for each friend in $\boldsymbol{D_{i,t}}$ each friend has exactly one of the scarce packets and does not have the other one. If such two packets are found then we XOR their information and we transmit them to the users. Upon receiving them, the users can XOR this information with the available packet, in this method being able to extract the unavailable packet from the information.
3. If no such two packets are found we try the same strategy with three scarcely available packets. For this case we are looking at finding three packets such that for each user in $\boldsymbol{D_{i,t}}$, he has two of the three packets available to him but not the third.
4. If Step 3 also fails, then we try to find 2 packets for which we can implement Step 2 on the largest subset of users from $\boldsymbol{D_{i,t}}$. (i.e. we try to find the largest subset of $\boldsymbol{D_{i,t}}$ for which Step 2 is possible.)

We note that in Step 4 we are looking at the largest number of users to which Step 2 can be implemented. The reason for this is that trying to do this for Step 3 (i.e. when we are working with 3 packets) can be overly complicated.

E. The Policy for Dropping Friends and Peer Selection

Since the information about the nodes distribution is not centrally available and the position of the nodes is constantly changing, one of the major difficulties in finding efficient information propagation techniques is to determine the best network topology to support it.

In order to address this problem we propose a network discovery technique which will work through periodic peer dropping and connection. This technique can be described as follows:

1. Let us define $\boldsymbol{S_{i,n,t}}$ as a quantity, where each user i ranks each of its peers j in $\boldsymbol{N_{i,t}}$ based on the number of packets j has sent to him since the last time i dropped a peer.
2. Periodically i decide to drop one of the $\boldsymbol{N_{i,t}}$ peers who has the lowest $\boldsymbol{S_{i,n,t}}$. At that time i picks one of the peers which: (i) has at least a certain percent of the file (usually 5 – 10 %) and (ii) has the lowest rank.

After dropping a peer the user searches for new friends which he may be able to connect to within its proximity. Based on a set of potentially new friends, i picks up enough new friends randomly as to fill up his pool of $\boldsymbol{N_{i,t}}$ friends to be equal to d. If not enough new peers are found to fill up the quota of d, then i picks up all of them. We define the above algorithm as a Rank Based Peer Selection (RBPS) scheme.

The intuition behind this friend dropping and adding is that user i will drop friends that either (i) have no new information which may be desirable to exchange, (ii) has moved out of range, or (iii) does not have enough power to transmit. This type of policy of network discovery will ensure that the links which contribute to the dissemination of the most useful information will be held, maximizing the value of information while minimizing the power required to keep connection to too many users at one time.

To provide a theoretical and experimental comparison of the RBPS algorithm, we propose the scheme with Random Peer Selection (RPS). The RPS algorithm is identical to RBPS except employing random peering strategy. Instead of evaluating rank and contribution of peers, this scheme drops randomly chosen available peers. The reason we are comparing our scheme to the random peering scheme is that the randomized approach allows us to derive an analytical expression. Also, it is fully distributed algorithm requiring no centralized knowledge. We expect that the RPS scheme will show good average performance forming highly dynamic link connections. Specifically, we denote each rank based peer selection and random peer selection scheme with network coding as RBPSNC and RPSNC respectively. Schemes without network coding, we denote as RBPS and RPS. Simulation analyses were performed in order to observe the effect of network coding.

For each of the algorithm, we measure performance in the following way: we measure the average round (time) to deliver all data files to all peers. This is to characterize the overall file sharing performance among all users. In addition, we measure the average switching time for reestablishing links to new peers for each policy. Due to power and resource constraints, it is recommended to minimize the number of switching, while maintaining good file sharing performance. Finally, we measure how fast a peer becomes a seed and can speed up the distribution process.

F. Analysis

We derived an expression to approximate the average number of rounds to distribute entire files to all users for uncoded and random peer selection case. This expression is based on some rough approximations, but seem to provide us answers which are very close to the simulation results. We assumed that the number of servers is one. Let us define, $E[T_{Comp}]$, as the average number of rounds required to complete entire file transmissions to all peers which can be written as

$$E[T_{Comp}] = E[\text{\# of messages that have to be sent to all peers}] / E[\text{\# of message transmitted in each round}]. \quad (1)$$

The denominator in (1) can be written as follows,

E[# of messages transmitted for each round]

$$\cong 1 \cdot d + (|S|-1)\sum_{i=0}^{d-1}(d-i)\cdot\frac{\binom{|S|-2}{d-i}}{\binom{|S|-1}{d}} \quad (2)$$

$$= d + \sum_{i=0}^{d-1}\frac{d!(|S|-1-d)!}{(d-i-1)!(|S|-1-(d+1-i))!},$$

where d is a number of peers and S is the number of all nodes in a network. The numerator in (1) is

E[# of messages that have to be sent to all peers] = $|\boldsymbol{F}|(n-1)$, where $|\boldsymbol{F}|$ is the size of file.

Hence, finally we obtain

$$E[T_{Comp}] \cong \frac{|F|(n-1)}{d + \sum_{i=0}^{d-1}\frac{d!(|S|-1-d)!}{(d-i-1)!(|S|-1-(d+1-i))!}}. \quad (3)$$

This is an approximation for $E[T_{Comp}]$, since this assumes that each peer always has new information to send to its peers and does not explicitly capture the time varying information distribution of the completed peers. However, (3) provides a good average completion performance for all peers and will be evaluated and compared with the simulation runs in the next section.

IV. SIMULATION

The initial simulation was run with 1 seed and 11 leeches, where a seed has a file composed of 500 chunks and each leech has p = 10 percent of random number of chunks available in the beginning. Each node randomly chooses 2 peers at the beginning. We measured the total number of rounds, $E[T_{Comp}]$, to distribute a file to all peers. We ran 100 iterations in order to compute the average value of $E[T_{Comp}]$. For each simulation that we run, we collected the total number of both uncoded and coded transmissions. Furthermore, for each algorithm, we computed the total number of peer drop and connection occurrences in order to characterize the efficiency of algorithm. In every T= 10 rounds, we simulated that each peer will choose a peer drop based on $\boldsymbol{S}_{i,n,t}$, and he will choose another available peer with a uniform distribution. Following Table I. provides the simulation results.

TABLE I. PERFORMANCE OF DIFFERENT ALGORITHMS

	RPSNC	**RBPSNC**	**RPS**	**RBPS**
$E[T_{Comp}]$	238.38	215.64	242.41	230.49
Total number of transmissions	5611.2	5397	5736	5675
Total number of network coded transmissions	219.28	762	0	0
Total number of changing peers	$2nE[T_{Comp}]/T$ =571.2	143.28	$2nE[T_{Comp}]/T$ =581.78	128.42

The total number of changing peers for RPSNC and RPS scheme can be trivially calculated by $2nE[T_{Comp}]/T$, since peers are kept for a duration of T and totally new peers get selected at each T. As we can see from the Table I, RBPSNC scheme shows the best performance for all performance criteria, while RPS scheme shows the worst performance. We can observe that total number of transmission is reduced by the opportunistic use of XOR coding. We can see that RBPSNC scheme took more advantage on the opportunistic XOR coding than RPSNC. On the other hand, network coding does not improve much performance on random peer selection scheme. Combined network coding and node rank approach yields the smallest number of information distribution time and transmission cost. Further, we evaluated (3) with $|S|=12$ and $F=500$ and obtain $E[T_{Comp}] = 225$. We can observe that the derived expression well approximates the average finishing time of the random peering schemes as shown from Table I. The performance of RBPS strategy outperforms the values of (3) with a better peering strategy and the use of network coding.

In Fig. 2 we characterize the dynamic file completion behavior for each node using the different schemes. In this figure we portray a single instance of file completion behavior for each node using both RPSNC and RBPSNC schemes. The X axis is the number of rounds and the Y axis indicates the percentage of file completion of each peer as time progresses. Also, a line with a slope = (Percentage of file need)/$E[T_{Comp}]$ is plotted to show the average file distribution growth as time progresses.

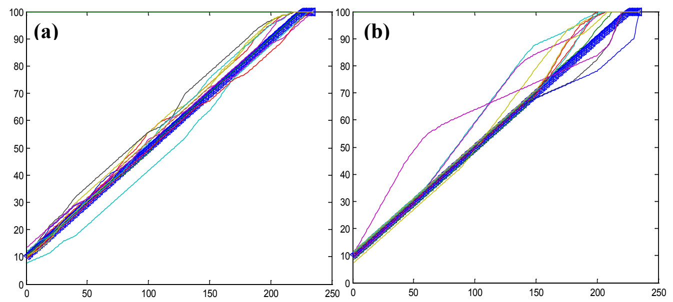

Figure2. File completion time in (a) Random Peer Selection with network coding (RPSNC) and (b) Rank Based Peer Selection with network coding (RBPSNC)

Some of peers in rank based algorithm completed a file reception around $T = 200$ rounds and rapidly assisted in helping other nodes complete the file. In RPSNC case most of nodes completed file reception almost at the same time. The rank based peer selection strategy exhibits more dynamics in the information sharing. The RPS and RBPS scheme exhibit the similar file completion behavior due to the same peering strategies as shown in Fig 2. (a) and (b) respectively. This is omitted to save space.

Also, we applied the different link failure probability to show the variability of link conditions. We performed the same simulation with 10 percent random node failure to evaluate the resilience of the algorithm. The results are provided in Table II.

TABLE II. PERFORMANCE OF DIFFERENT ALGORITHMS WITH LINK FAILURE PROB. = 0.1

	RPSNC	**RBPSNC**	**RPS**	**RBPS**
$E[T_{Comp}]$	255.43	247.34	260.4064	257.72
Total number of transmissions	5395	5494	5562.2	5721
Total number of network coded transmissions	258.18	514	0	0
Total number of changing peers	$2n\text{E}[T_{Comp}]/T$ =613.03	224.43	$2n\text{E}[T_{Comp}]/T$ =624.97	260.72

From Table II, we can observe that $E[T_{Comp}]$ was increased due to link disruptions. As we can see from this table, the proposed RBPSNC scheme is robust for abrupt link disruptions and it is more prone to distribute the file pieces more reliably when compared to other schemes, given that there is a small overhead in peer dropping and connection setup. Surprisingly, the RPSNC shows the least number of total transmissions needed to complete file transmissions among peers when there is a large number of link failures. Although this metric may not be as important as the file completion time, this may show there may be some benefits to the random peering when the network link disturbances are high.

Initial simulation results show that the combined network coding and rank-based peer selection improves the average content distribution time compared to random peer selection strategies as shown in Table I and II. The opportunistic network coding shows some improvement when it is used with rank based peer selection strategy.

V. CONCLUSION

The proposed protocol shows promising results on efficient content distribution in highly dynamic network environments. Our algorithm can be integrated as a multi-tier network component to provide efficient file sharing services in the context of Disruption Tolerant Networking technologies. Currently, we are investigating node interactions and information distributions using random graph and spectral theory to obtain deeper understanding on the network dynamics to improve file distribution performance.

ACKNOWLEDGMENT

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology. The authors would like to thank Loren Clare, Matt Klimesh, and Kar-Ming Cheung for providing helpful insights.

REFERENCES

[1] B. Cohen, “Incentives build robustness in bittorrent,” in *P2P Economics Workshop*, Berkeley, CA, 2003.

[2] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network Information Flow," *IEEE Transactions on Information Theory*, IT-46, pp. 1204-1216, July 2000.

[3] C. Gkantsidis and P. Rodriguez, “Network coding for large scale content distribution,” in *IEEE Infocom*, FL, 2005.

[4] C. Gkantsidis, J. Miller, and P. Rodriguez, “Comprehensive view of a live network coding P2P system,” in *ACM SIGCOMM/USENIX Internet Measurement Conference* (*IMC*), Brazil, 2006.

[5] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft "XORs In The Air: Practical Wireless Network Coding," in *ACM SIGCOMM*, 2006.

[6] C. Fragouli, J. Widmer, and J.-Y. LeBoudec, "Network coding: an instant primer," in *ACM Computer Communication Review*, 2006.

[7] J.-S. Park, D. S. Lun, Y. Yi and M. Gerla, "CodeCast: a network coding based ad hoc multicast protocol," in IEEE Wireless Communications, 2006..

[8] D. Nguyen, T. Nguyen, and B. Bose, "Wireless Broadcast Using Network Coding," in *IEEE NetCod Workshop*, 2007.

[9] D. Chiu, R. Yeung, J. Huang and B. Fan, "Can Network Coding Help in P2P Networks," in *Second Workshop of Network Coding*, in conjunction with *WiOpt*, April 2006.