# Reliable Multicast using Rateless Forward Error Correction and Orthogonal-Scheduling in MANET

Jeyasekar A[1], Subhajit Saha[2],
[1] SRM University, Chennai, Tamilnadu, India, ajeyasekar@yahoo.com
[2] Former M.Tech Student, SRM University, Chennai, Tamilnadu, India, warrior_prince_one@yahoo.co.in.

**Abstract** – *Collision free transaction and error correction system always a challenging issue in mobile ad hoc networks because of its innate characteristics of mobility, link failures and erroneous channels. An effective scheduling methods and error correction methods will obviously improve the Quality of Service of MANET. In collision free transmission, it assured that transmission takes place without any collision with other nodes. Even though there is a collision free transmission, some time the packets are dropped at receiver due to bit error or packet errors of received packets. This paper provides a way to achieve a collision free and reliable multicast transaction in MANET. The deterministic scheduling approach instead of probabilistic approach for channel access using orthogonal array method provides collision free transaction. A rateless forward error correction as an acknowledgement scheme using Luby Transform Code is used to provide a reliable multicast transaction. To provide greater performance and reliability the neighbors' node of the multicast group send a message back to the transmitter to stop sending the encoded packets for the receiver node which has correctly decoded the message. The deterministic Schedule and Rateless Forward Error Correction for reliable multicast is simulated using NS2 simulator. The results show that there is observable improvement in throughput.*

***Keywords***: *MANET, Rateless Forward Error Correction, Orthogonal Array scheduling, Cover-free family, Multicast*

## I. Introduction

Mobile ad hoc network uses Medium Access Control (MAC) protocols to share the channel with all nodes within the transmission range which leads to collision sometimes. A scheduled approach to channel access provides deterministic delay guarantees and avoids the collision. But scheduling in MANET is very difficult because there is no fixed infrastructure and centralized control [8].

Many scheduling schemes have been proposed to exploit spatial reuse of channel. These schemes are based on two approaches named Topology-dependent and Topology-transparent. In topology-dependent approach, the access scheme alternates between contention phase and collision-free phase. During the contention phase, neighbor information is collected and during the collision-free phase, the mobile nodes transmit according to a schedule constructed using the neighbor information. In contrast, the idea in topology-transparent protocols is to design schedules that are independent of the detailed network topology. Specifically, the schedules do not depend on the identity of a node's neighbors, but rather on how many of them are transmitting. This approach does not use any information about who is a neighbor of whom. The schedules for topology-transparent access protocols depend on two design parameters: $N$, the number of nodes in the network, and $D_{max}$, the maximum node degree. The topology-transparent transmission schedule is prepared with the help of orthogonal array [11][12].

Even though collision is prevented by using topology transparent scheduling scheme, sometimes, the packets are dropped at receiver end due to bit errors and packet errors of transmitted message. The lost packets are recovered by using the packet loss recovery schemes which is classified into two categories known as Transmission based error recovery scheme and Coding based error recovery scheme. In the first scheme, the lost packets are retransmitted by sender once again. But retransmission is inefficient for multicast communication. In the later scheme, codes are generated from original data packets and the sender transmits both the original data packets and redundant code. This code helps the receiver to recover the lost data completely if the number of lost packets is less than to redundant code. This scheme is called forward error correction method.

Code based error recovery scheme uses digital fountain paradigm. A digital fountain (DF) can encode and transmit an unlimited number of data packets until every user gets enough information to guarantee correct decoding. The encoder is represented by fountain that

*Manuscript received on August 2016, revised on September 2016*

produces a potentially infinite amount of water drops. Every drop represents an encoded packet, which has the same size as a source symbol. At the receiver side, the decoder is represented by bucket. It collects water drops that have spurted from the fountain until the bucket is full. Finally, the decoder is able to recover the original information, independently of which drops it has collected. This paradigm is useful for data communication over a network subject to packet reassure. The design of two practical DF codes is Luby transform code (LT code) and Raptor codes. The LT code as rateless FEC doesn't require knowledge of the loss rate on the channel. It permits fast encoding and decoding algorithms. The decoding is successful in recovering the original message once an amount of data marginally larger than the original data is received [11][7].

In this paper, we focus on topology-transparent scheduling and error recovery scheme for multicast. OA is used to develop topology-transparent scheduling and the LT code is used for packet level error recovery scheme.

## II. Motivation

Violet et al [11] suggest a rateless FEC for topology-transparent scheduling by using orthogonal array (OA) and LT code in unicast transmission and show that there is no observable negative effect on throughput but a small impact on delay for unicast transmission. Erasure code proposed in [11][12][2] provides reliability for one-to-one delivery. The advantage of using erasure codes in one-to-one data delivery is that it makes the designing of flow and congestion control mechanism easier. For the one-to-many data delivery, the feedback to the sender needs to be limited. If erasure codes are powerful enough, then a single sender can potentially be used to reliably deliver data efficiently to a large number of constant receivers without feedback [7]. Therefore we focus on topology-transparent scheduling scheme with rateless forward error correction scheme for multicast in MANET

## III. Related Works

Ouyang X Hong et el classified the reliable multicast protocols into three categories according to the error recovery mechanisms named ARQ-based protocol, Gossip-based protocol and ARQ-FEC based protocol. In ARQ-based protocols and gossip-based protocols uses retransmission from sender and retransmission form local nodes rather than from sender respectively. But in ARQ-FEC based protocols [9], the sender transmits partial data from group of data along with redundant code to receivers rather than transmits all redundant data for each group of data. When a receiver detects a packet loss, it informs the sender to transmit more redundant data. However the feedback implosion problem, scalability problem are the drawback of this protocol [9]. Therefore Violet R Syrotiuk et el proposed Rateless Forward Error Correction (RFEC) which uses LT code and Topology-Transparent scheduling using OA. It shows no observable negative effect on throughput of unicast transmission. Therefore we use these two techniques ie LT code and OA for multicast transmission in MANET.

### *III.1. Orthogonal Array*

Let $V$ be a set of $v$ symbols denoted by $0, 1, \ldots, v-1$. Let $k \times v^t$ array $A$ with entries from $V$ is an *orthogonal array* with $v$ levels and strength $t$ (for some $t$ in the range $0 \le t \le k$) if every $t \times v^t$ subarray of $A$ contains each $t$-tuple based on $V$ exactly once as a column. We denote such an array by $OA(t, k, v)$.

Table 1 shows $OA(2, 4, 4)$, an example of an orthogonal array of strength $t = 2$ with $v = 4$ levels, i.e. $V = \{0, 1, 2, 3\}$. Select any two rows, for example third and the fourth. Each of the sixteen ordered pairs $(x, y)$, $x, y \in V$ appears the same number of times, once in this example OA(2,4,4) in the third and fourth row. The upper bound of the transmission schedule of the OA is the number of nodes in the network. Each column, which is called a *codeword*, gives rise to a transmission schedule. Each codeword intersects every other in fewer than $t$ positions. For example, the first and the eighth column intersect in no positions, while the first and the second column intersect in the first position (row) with zero symbol.

**Table 1: Orthogonal array O**$A(2, 4, 4)$**.**

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 |
| 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 |
| 0 | 1 | 2 | 3 | 1 | 0 | 3 | 2 | 2 | 3 | 0 | 1 | 3 | 2 | 1 | 0 |
| 0 | 1 | 3 | 2 | 3 | 2 | 0 | 1 | 2 | 3 | 1 | 0 | 1 | 0 | 2 | 3 |

The importance of this intersection property is as follows. Select any codeword. Since any of the other codewords can intersect it in at most $t-1$ positions, any collection of $D$ other codewords has the property that our given codeword differs from all of these $D$ in at least $k - D(t-1)$ positions. Provided this difference is positive, the codeword therefore contains at least one symbol appearing in a position, not occurring in any of the $D$ codewords in the same position. In our application this means that at least one collision-free slot to each neighbour exists when a node has at most $D$ neighbors. Thus as long as the number of neighbors are bounded by $D$, the delay to reach each neighbor is bounded, even when each neighbor is transmitting. Clearly, the orthogonal array gives a $D$ cover-free family [11].

### III.2. *Rateless Forward Error Correction*

FEC technique enables a receiver to correct errors/losses without further interactions with the sender. FEC potentially provides a low latency method for correcting packet losses. In contrast with other error recovery mechanisms such as ARQ, FEC does not require feedback or an upstream channel. It can immediately recover lost packets without having to wait for feedback from the source. This is convenient for long latency links and communication connections without an upstream channel such as satellite links. End-to End delay for FEC based loss recovery mechanism is smaller than that for retransmission based loss recovery mechanism. There are two types of FEC techniques i.e. Byte level FEC and Packet level FEC. In byte level FEC, a symbol is a packet. Packet level FEC produces n redundant packets from k original packets. In this paper, packet level FEC is used.

An (n, *k*) block erasure code converts *k* source data into a group of n coded data. Such that any *k* of the n coded data could be used to reconstruct the original source data. Usually, FEC codes are able to correct the errors and erasures in an unknown position where as an erasure is a corrupted symbol in a known position [7].

LT code as RFEC scheme does not require knowledge of the loss rate on the channel. It is assumed that the bit errors or packet errors are due to channel condition and independent of collisions. But collisions are the only cause of erasures [11]. Hence LT code with topology transparent transmission schedule may be the best selection for reliable communication because topology transparent environment is used for the allocation of time slots for each node to transmit rather than channels to nodes. The parameters that characterize the LT process are: 1) the size $l$ of each block or encoding symbol; 2) the degree $\alpha$ of the encoding symbol; 3) the number of blocks $p$ into which a message of $M$ bytes is divided; and 4) the number of encoding symbols needed to completely recover the message, denoted by $(p + \delta)$. A message of size $M$ bytes is fragmented into $p = [M / l]$ blocks; these blocks are numbered sequentially. A degree $\alpha$ is chosen at random from a soliton distribution. Now $\alpha$ distinct blocks are chosen from the message uniformly at random. The value of the encoding symbol is the exclusive-or (modulo 2 sum) of these $\alpha$ blocks.

In practice, 5% more than the original message data is needed to reconstruct an exact copy of it by the receiver. It does not matter which of the encoding symbols are received, or in what order they are received. All the encoding symbols are of equal value in reconstructing the message. Indeed, although not our primary concern here, the reception of symbols need not be by any agreed upon route or by a single route at all. We do not explore the possible advantages in terms of simultaneous multi-path routing [11].

# IV. Proposed System

## IV.1. *Network Model*

A graph G = (V, E) where $V = \{v_1, v_2, \ldots\ldots., v_N\}$ represents the nodes, and E represents the communication links; N= |V| is used to model the MANET. An omnidirectional antenna is attached with each node with transmission range of circle of radius r. there is an edge $\{v_i,v_j\}$  E between nodes $v_i$ and $v_j$ if the distance separating the nodes is within the transmission range. If $v_i$ is adjacent to $v_j$ then $v_j$ is a neighbor of $v_i$. The maximum node degree of a MANET G is $D=\max^{N}_{i=1} d(v_i)$, where $d(v_i)$ is the degree of $v_i$. The transmission at each mode is half –duplex; it cannot transmit and receive at the same time. As a result, some strategy is required to inform the transmitter of the outcome of its transmission.

Time is divided into discrete unit called slots. A schedule $S_i$ for node $v_i$ is a binary vector $s_1,s_2,\ldots\ldots,s_N$ with one element corresponding to a transmission decision for each slot in the frame. A node $v_i$ with schedule $S_i$ may transmit in slot k whenever $s_j = 1$ and $k \equiv j \pmod N$; otherwise the node is silent (and could received).

In designing a topology-transparent transmission schedule with parameters N and D we are interested in the following combinatorial property. For each node, we must guarantee that if a node $v_i$ has at most D neighbors its schedule $S_i$ guarantees a collision-free transmission to each neighbor.

Let us treat each schedule $S_i$ as a subset $T_i$ on {1, 2, ....., N}; $T_i$ is the characteristic set of $S_i$. Now the combinatorial problem asks for each node $v_i$ to be assigned a subset $Y_i$ with the property that the union of D or fewer other subsets cannot contain $T_i$. Expressed mathematically, if D is a set of at most D of the sets $\{T_i\}$, and $T_i \notin D$, then $(\cup_{T\in D} T) \not\supseteq T_i$. This is precisely a D cover-free family. These are equivalent to D disjunct matrices [19] and to superimposed codes of order D [11][2].

## *IV.2. Sender Side Algorithm*

1: Convert the given message into binary format (B)
2: Divide B into n blocks of size 1 i.e. $b_1, b_2, b_3, \ldots\ldots.., b_n$
3: Select a code word from the constructed orthogonal array
4: Repeat the following
  a) Select a degree d from soliton distribution
  b) Select uniformly at random of d blocks from B
c) Encode the message by performing XOR operating on the selected blocks
5: Append degree d and selected block numbers to the resultant encoded message
6: Send the encoded message to destination nodes according to the codeword until an acknowledgement from the receiver is received.

## *IV.3. Receiver Side Algorithm*

1. For each received encoded packet $M_i$,
  1.1 Extract the degree and block numbers and store it
  1.2 If degree d=1 with block no. $b_i$,
    a) Store the message as decoded message in the block no. $b_i$
      Else
  1.3 If degree d=2 with block nos. $b_i$ and $b_j$
    a) Search for any of the recovered blocks either $b_i$ or $b_j$
    b) Perform XOR operation to $M_i$ with recovered block to recover the second block ($b_i$ or $b_j$)
2. Repeat the above process for other degrees until all blocks are recovered
3. Send an acknowledgement to the sender saying that the message is received fully

## V. Simulation and Performance Analysis

We use network simulator ns-2 version 3.10. A total of N=25 nodes is generated, distributed over a simulation area and connected to their peers via a shared 11 Mbps wireless interface.

| Simulation Parameter | Value |
|---|---|
| *Simulation* | *NS-2 Version 3.10* |
| *Simulation Area* | *670 x 670 $m^2$* |
| *Number of Nodes* | *25* |
| *Packet Length* | *1 Byte* |
| *Mobility Model* | *Steady state random waypoint* |
| *Wireless Interface* | *11 Mbps* |
| *Transmission Range* | *250 m* |
| *Propagation Model* | *Two ray ground* |
| *Routing Protocol* | *Ad-hoc distance vector routing(AODV)* |
| *Node Speed* | *Static and 20 m/s* |
| *Simulation time* | *30 s* |

For analyzing the throughput of the system we have collected the data of number of packets sent from the sender and the number of packets drops and losses through the trace graph. Then we have collected the data for the number of successful transmission for each of the neighbor nodes in the multicast environment for different sources which are sending the data according to the orthogonal array that has been constructed so that collision could be avoided. So the throughput general formula for calculating the performance of the system is:

*Successful packets = total packets sent – packets dropped & lost* (1)

*Throughput = (Successful packets / total packets sent) x 100* (2)

Here we have the performance analysis for two sources to generalize the performance. These two sources are having three common nodes where they are multicasting the data according to topology-transparent scheduling to avoid collision.

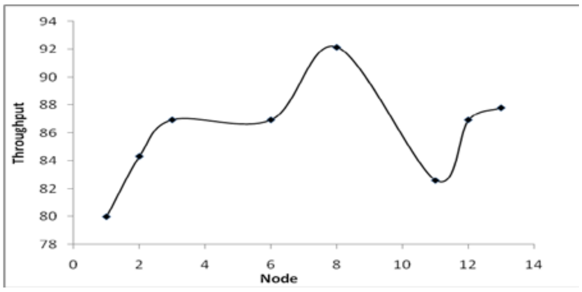

*Fig. 1 Throughput Analysis for Source 7*

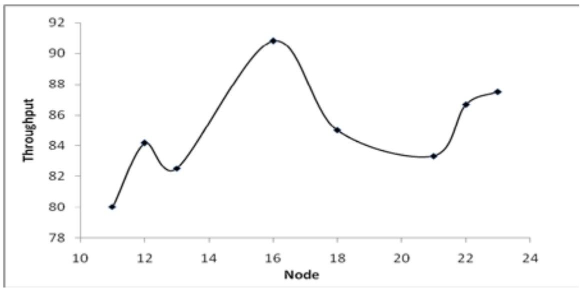

*Fig. 2 Throughput Analysis for Source 17*

From the Fig 1, we can see that the *throughput* of the system is always between 80% to 92%. If there is a more packet are dropped, the *LT Code* keep sending enough packets to reconstruct to original message so it is compensating the dropping of the packets which provides the higher throughput. The dropping of the packets only due to error in the packets, there is no packets dropping due to collision. Here we have compared the system with a normal system where no collision avoidance techniques have been introduced. The collision has been avoided on the node number 11, 12 & 13.So we have taken the data for these three nodes.

From the Fig 2, the curvature of the lines is same, which implies that there is a fixed improvement of the performance in the system developed. So from the graphs we found that in our system there is an improvement of around 6.5% in *throughput*

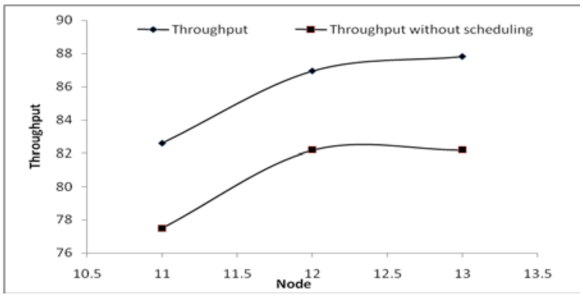

*Fig. 3Throughput Comparison Analysis for Source 7*

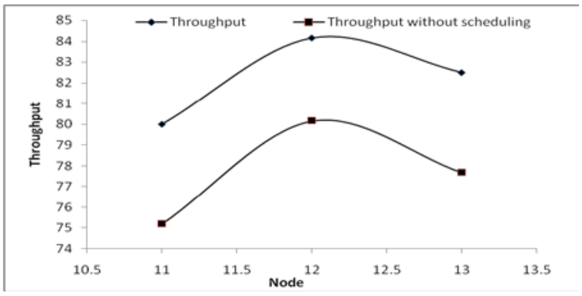

*Fig. 4Throughput Comparison Analysis for Source 17*

To show the system's performance in different *loss rate* we have collected the data in different numbers of *neighbors*. From the Fig. 5, it is evident that whatever loss rate it could be but the system will always give the performance *throughput* above 80%. Here the throughput with different loss rate and degree is in between 80% to 96%. From the graph we can see that with higher degree whatever be the loss rate it gives better throughput. With degree 6, 7 & 8 the throughput is around 90% in all loss rates.

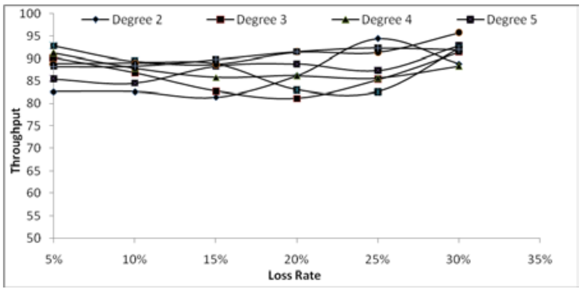

*Fig. 5) Throughput in different Loss Rate and Degree*

## VI. Conclusion and Future Work

Violet R Syrotiuk et al. showed the topology transparent medium access control protocol corresponds to an orthogonal array using LT code and it was simulated for reliable unicast transmission. In this work, we proposed a scheme for multicast transmission with topology transparent scheduling and rateless forward error correction scheme because TCP retransmission is in efficient for multicast transmission. It improves the traditional scheme with respect to throughput, reliability and packet loss recovery. The same system can be made using *Raptor Code* which is more efficient Forward error Correction method which is expected to improve the throughput more.

## References

[1] A. S. Hedayat, N. J. A. Sloane, and J. Stufken, Orthogonal Arrays, Theory and Applications. New York: Springer-Verlag, 1999.

[2] C. J. Colbourn, A. C. H. Ling, and V. R. Syrotiuk, "Cover-free families and topology-transparent scheduling for MANETs," *Designs, Codes, and Cryptography*, Vol. 32, No. 1, pp. 35–65, May 2004.

[3] D.J.C. MacKay. Fountain Codes. IEE Proc.-Commun., Vol. 152, No. 6, Dec 2005

[4] Esa Hyytiä,Tuomas Tirronen, Jorma Virtamo. Optimizing the Degree Distribution of LT Codes with an Importance Sampling Approach.

[5] J.-H. Ju and V. O. K. Li, "An optimal topology-transparent scheduling method in multihop packet radio networks," *IEEE/ACM Trans. Networking*, Vol. 6, No. 3, pp. 298–306, Jun. 1998.

[6] J.W. Byers, M. Luby, M. Mitzenmacher, A. Rege. A Digital Fountain Approach to Reliable Distribution of Bulk Data. Proceedings of *ACM SIGCOMM '98*, pp.56–67, Vancouver, Sep 1998.

[7] M. G. Luby, "LT codes," in Proc. 43rd Symp. Foundations of Computer Science (FOCS'02), Nov. 2002, pp. 271–280.

[8] Muhammad Hisyam Lee,Mazalan Sarahintu" Performance analysis of Dynamic Source Routing protocol for Ad Hoc Networks Based on Taguchi's Method", MATEMATIKA, 2008, Vol. 24, No. 2, pp. 199–209.

[9] Ouyang X Hong, Y.yi, " A Compariosion of reliable multicast protocols for Mobile Ad hoc networks." in proceeding of IEEE Southern east Conference,pp344, Apr 2005.

[10] The network simulator: ns-2. Univ. California, Berkeley [Online]. Available: www.isi.edu/nsname/ns/.

[11] Violet R. Syrotiuk, *Member*, *IEEE*, Charles J. Colbourn, and Sruthi Yellamraju. Rateless Forward Error Correction For Topology-Transparent Scheduling**.** IEEE/ACM Transactions on Networking, Vol. I6, No, 2, April 2008

[12] V. R. Syrotiuk, C. J. Colbourn, and A. C. H. Ling, "Topology-transparent scheduling for MANETs using orthogonal arrays," in Proc. DIAL-M/POMC JointWorkshop on Foundations of Mobile Computing, Sep. 2003, pp. 43–49.