POSET: Network Coding Multicast in Application Layer

T. Ruso and C. Chellappan

***Abstract*—Application Layer Multicast (ALM) protocols suffer from delay, throughput and security. To address these all issues, network coding techniques has been introduced. Existing protocols are using either mesh or tree kind of topology structure for data distribution. In these topology structures network coding is too complex. To address this problem, a novel POSET protocol stack has been proposed. It consists of three protocols such as control protocol, poset structure protocol and data distribution protocol. Control protocol controls join, leave and prune operations. Poset protocol forms poset structure using lattices property and distribution protocol consist of finite filed encoding, intermediate encoding and finite field decoding modules. The results have been compared with existing Nice protocol and increases 8.25% of throughput and decreases 33% of delay.**

***Index Terms*—Delay, mesh-tree, security, throughput.**

I. Introduction

Many network applications, such as video conferencing, telemedicine, remote classrooms, distributed interactive gaming, software upgrading and distributed database replication, requires multicast communication. Stephen Deering first introduced multicast in [1], a lot of work have been done on multicast routing protocols. Most works focusing on extending multicast routing protocols. Usually this routing protocol constructs a multicast spanning tree from source to all the group members. The nodes (switches or routers) in the network replicate packets to reach multiple receivers such that message is sent only once. To implement multicast in IP layer, routing protocols such as Open Shortest Path First Protocol (OSPF) [2], Distance Vector Multicast Routing Protocols (DVMRP) [3], Internet Group Management Protocol (IGMP) [4] and Protocol Independent Multicast-Sparse Mode (PIM-SM) [5] and Protocol Independent Multicast-Dense Mode (PIM-DM) [6] are introduced.

Due to the limited commercial deployment of IP Multicast, there is a need for a more robust and secure way of transmission. Hence the entire IP Multicast functionalities are shifted into the application layer. ALM creates an overlay network rather than an IP layer network. This has two major advantages: first, network nodes are in an overlay network, as opposed to lower-layer network elements such as routers, are end systems and have capabilities far beyond basic operations of storing and forwarding.

Second, the topology of an overlay network can be manipulated wilfully to suit one's purposes since it resides on top of a densely connected IP-layer network. The links between nodes can be dynamically created or torn down to construct topologies that are conducive to better network performance.

There are many ALM protocols have been introduced, for example like NICE, ZIGZAG, OMNI, etc., these are not changing the existing network infrastructure; instead, they implement multicasting forwarding functionality exclusively at end-hosts. Thus it can deal with logically created nodes instead of physical nodes [7].

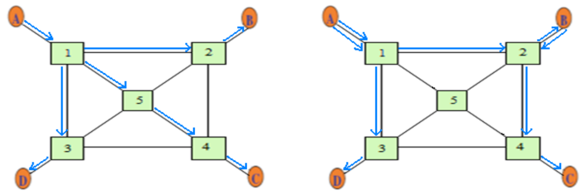

Fig 1. a) IP multicast b) Application layer multicast.

The figure above (Fig. 1. [3]) shows the basic difference between network layer multicasting and application layer multicasting. ALM protocols suffer from the issues like delay stretch, stress, throughput, reliability, Group management, Network management and security [7]. Stress is defined as the number of identical copies of a same data packet traversing the link and stretch is defined as the ratio of the delay along the overlay to the delay of path from source to destination. Lastly, ALM overlay is infamous for being unstable as a result of failure of end hosts or the wilful exit of nodes.

To solve the ALM issues, network coding technique has been introduced [8]. It is mainly used to improve delay, network throughput, resilience to attacks and eavesdropping. It allows Intermediate nodes to encode several packets into a single packet and allows Maximum data transfer rate in single source multicast network.

In this paper we address two issues such as delay and throughput by applying the poset architectural data distribution with network coding technique. The major objectives of this paper are: minimize the link stress on the network topology and maximize the throughput for hard real time application.

A partially ordered set (or poset) formalizes and generalizes the intuitive concept of an ordering, sequencing, or arrangement of the elements of a set. A (non-strict) partial order is a binary relation "≤" over a set P which is antisymmetric, transitive, and reflexive, It consists of a set together with the binary relation that indicates that, for certain pairs of elements in the set, one of the elements precedes the other. This mesh-poset topology structure reduces end-end delay and increases throughput and provides the security.

The rest of the paper is structured as follows: In Section II,

Manuscript received May 6, 2013; revised August 5, 2013.
T. Ruso and C. Chellappan are with Department of Computer Science and Engineering, Anna University, Chennai-25, India (e-mail: racingruso@gmail.com, drcc@annauniv.edu).

DOI: 10.7763/IJEEEE.2014.V4.298

describes the related works and explain how different ALM protocols and network coding techniques are working. In Section III, describes proposed the mesh-poset protocol architecture. In Section IV, described modules in proposed system. In section V, simulation setup has been described, in Section VI, discussed performance evaluation and in Section VII, we conclude our work.

II. Literature Survey

ALM is an implementation of multicasting functionality as an application layer instead of a network layer. While IP Multicast is implemented by network nodes (i.e. Routers) and avoids multiple copies of the same packet on the same link as well as possibly constructing optimal trees, ALM is implemented by application nodes (either end systems or proxies) and results in multiple copies of the same packet on the same link as well as typically constructing non-optimal trees. To solve IP multicast issues like routing, group management, address allocations, authorization and security, QoS, scalability, so many ALM protocol has been introduced.

TABLE I: Comparison of Different ALM Protocols

ALM Protocol	Topology Structure	Structure Refinement	Node Joins At	Control Overhead
NARADA	Mesh	Periodically	Random join	-
CAN	Mesh	No	Closest node	*O(d)*
P2P	Mesh	At every join	Random join	-
HMTP	Tree	Periodically	Designated member of the island	-
LAPTOP	Tree	No	Not specified	-
TAG	Tree	No	Root	*O(k(log N))*
OMNI	Tree	Periodically	Source	O(log N)
ZIGZAG	Tree	Periodically	Cluster head	O(K*log N) or O(K)
NICE	Tree	At every join	Cluster head	O(log N)
Bin Casting	Tree	At every join	Root	-

From Table I, we understood that NARADA, CAN, P2P protocols are mesh based and HMTP, LAPTOP, TAG, OMNI, ZIGZAG, NICE, BinCast are tree based protocols. These protocols are only changing the topology structure. Issues of ALM [9] are Delay stretch, Bandwidth, Throughput, Reliability, Security, Group Management Network Management. To address the ALM issues, Network technique has been introduced [10].

Network Coding [11] generalizes tradition store-and-forward routing techniques by allowing intermediate nodes in networks to encode several received packets into a single coded packet before forwarding. Network coding is done using Coding matrix as follows:

$$Y=[C]\,[X]$$

In Network coding [12], [13] the nodes in a network are allowed to encode the information received from the input links. Specifically, a collection of subsets of links is given, and a wire tapper is allowed to access a link from a collection of links without being able to obtain any information about the message transmitted.

A distributed random linear network coding approach [14], for transmission and compression of information in general multisource multicast networks is presented. Network nodes independently and randomly select linear mappings from inputs onto output links over some field. Random linear coding performs compression when necessary in a network. Benefits of this approach are decentralized operation and robustness to network changes or link failures. Simulation based approach is implemented using steiner tree algorithm.

Fig. 2 illustrates a distributed random linear network coding. X_1 and X_2 are the source processes being multicast to the receivers and the coefficients are randomly chosen elements of a finite field. The label on each link represents the process being transmitted on the link.

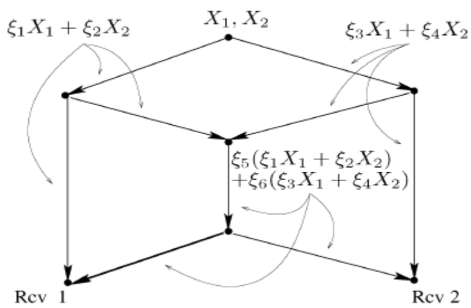

Fig. 2. Distributed random linear network coding.

File distribution via network coding [14] has received a lot of attention lately. However, direct application of network coding may have security problems. In particular, attackers can inject "faked" packets into the file distribution process to slow down the information dispersal or even deplete the network resource. Therefore, content verification is an important and practical issue when network coding is employed. The Content verification done using 'HASH AND SIGN TECHNIQUE'.Given original data X and a collision-free hash function h, a hash value h(X) is computed, which is then signed by employing a digital signature scheme S with some signing key k, and the signature Sk(h(X)) is published. Problem 1: The total size of the hash values is proportional to the number of blocks, which could be very large. Problem 2: The cryptographic hash function proposed is computationally expensive.

The original m information flows on source side are composed as one group and encoded into n information flows ($n>=m$) with equal size which are re-encoded and forwarded by the intermediate node [15]. The original information flow could be acquired through the decoding algorithm after destination receives all the packets.

Source side encoding: When information is needed on the source side m numbers g_1 ,g_2…. g_n from a finte field F is randomly selected and they are linearly encoded as below:

Intermediate node re-encoding: Intermediate node stores the encoded message and re-encodes the message with same group label. Suppose the intermediate node receives K encoding information Y_1, Y_2,… Yk, these are re-encoded using encoding factor hi_1, hi_2,… *hik* where *h* is the inner product of matrix g and encoding vector matrix *k*.

Destination side decoding: The m original source information is obtained at the receiver node by finding the product of the above output and encoding matrix inverse. If the received encoding is less than m the information feedback mechanism could inform the upstream node to re-encode and re-forward until the information could be recovered by the destination node.

III. POSET Protocol Stack

In Fig. 3, POSET protocol stack consist of three major blocks such as Control Protocol, Multicast Tree construction and Multicast content distribution.

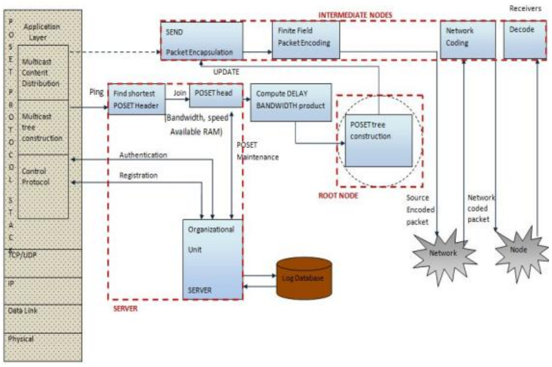

Fig. 3. POSET architectural diagram.

A. *Control Protocol*

There are three control messages: JOIN, LEAVE and PRUNE. For each node in layer i, messages with adjacent parent node of the multicast tree is periodically exchanged.

B. *Multicast Tree/Poset Construction*

Every cluster has a representing node called the POSET head which is optimally selected using the delay x bandwidth product. This POSET head looks after the POSET construction based on the product so that weak nodes are not over headed. The tree also varies based on the number of clients which is discussed in detail later.

```
Poset architecture construction algorithm.
For every registered node
   Count← count+1
Endfor
If count <8          //ring topology
   For i=0..7
      node [i]←→node[i+1]
   endfor
   node[count]←→node[0]
else if count=8 || 16        //Lattice topology
   create poset
else
   create poset for 8
   create ring for the rest
```

C. *Multicast Content Distribution*

In data distribution, each registered node is aware of its neighbours. Data packets are sent from the source to the destination via the intermediate nodes in such a way that the following operations are performed.

First intermediate node performs a finite field encoding. Second intermediate node performs Network Coding as a way of mixing 2 packets using global encoding vector and is forwarded. Finally the receiving side distributor decodes the packets and broadcasts the data to all its receivers.

IV. Module Description

As clearly shown in Fig. 3, each and every participating node should register in the server before multicasting of information. The Processor and system configuration are analyzed. Collected Metrics are processed. POSET value is assigned. Nodes are arranged using POSET property. Clusters of nodes are formed. Data is then distributed. The modules in our system are listed below:

A. *Network Setup*

In this project, we deal here with cooperation, unreliable communication and highly dynamic environment. We aim at achieving robustness and tolerance to failure in a setting where agents can be redundant and communication is unreliable.

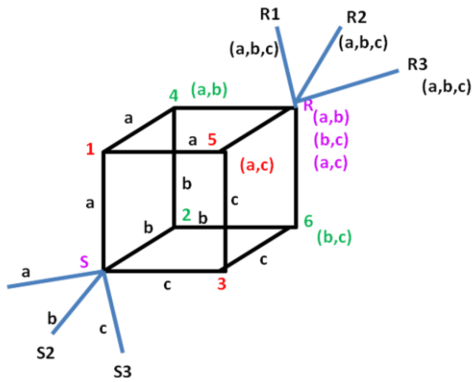

Fig. 4. POSET architecture。

Fig. 4, we create a mesh-POSET architecture wherein the participating nodes are arranged in the POSET based on the POSET property. The network is set up in such a way that different POSETs can interact with each other through a common server and each POSET has its own head which co-ordinates the intra-POSET communication.

1) *Buffer maintenance*

A region of memory is allocated for the buffer which is used to temporarily store the packets from the server while the preceding set of packets are being processed to the destination. The packets are released only after receiving an acknowledgement from the destinations. The buffer adjusts timing by implementing a queue algorithm in memory.

2) POSET property

The POSET property (i.e.) delay x bandwidth product in this case is calculated for every node. It indicates the maximum amount of data that can be handled at a given time. Hence the major advantage of setting up the network based on this property is that only the most efficient and capable (i.e. having highest delay x bandwidth product) node is assigned to do network coding (as it involves more computation). Every time the simulation runs, the POSET property value for each node changes and hence the tree constructs dynamically based on this value.

B. Dynamic Topology Creation

Since the client can give any number of nodes as input, the topology dynamically changes according to the input. For an input of: Less than 8, Ring topology is created Between 8 to 16, A POSET of 8 and Ring for the rest is created Greater than 16, Two POSETs are created

C. Finite Field Encoder

Random linear coding is adopted for network coding scheme. The information codes are grouped and are labelled. The POSET head receives multiple packets from different clusters and stored in a buffer. Based on the constraints of time slot and buffer size, the packets are lined for encoding.

Encoding is done by splitting the input text into suitable size and converting it into MATRIX format. This is then combined with the finite field to produce the encoded matrix. Encoding factor and the group label are also added into head information. As per the above Fig., node 0 does encoding.

$$E_i = m\sum j\text{-}1 \;\; F_{ij}A_j \quad i=1, 2, .., n \qquad (1)$$

D. Network Coder

In the POSET architecture, the encoded packets are multicasted to the connected nodes where network coding takes place. Two packets are collected in all the network coding areas and these are combined using a double-layered misplacement key which is shared only with the receiver.

E. Finite Field Decoder

The network coded packets are separated using the secretly shared key, decoded and then broadcasted to the intended destinations.

$$AJ = m\sum j\text{-}1 \;\; Y_{ij}F\text{-}1 \quad i=1, 2, .., n \qquad (2)$$

V. Simulation

Poset protocol is simulated by omnet++. The network topologies were developed using OverSim, which is a simulation framework for overlay and P2P networks. The simulator allows collecting data on sent, received packets and the network traffic per node.

Table II, shows the simulation parameters. In simulation set up, 500 nodes with different bandwidth, memory and processor speed, have been taken. The end to end delay is calculated while sending fixed size packet (1024 bytes). Delay and bandwidth metrics are used to calculate POSET value. Delay is measured by adding packet delivery delay, propagation delay and route delivery delay. It is typically measured in multiples or fractions of seconds. Delay may differ depending on the type of processing the node does.

TABLE II. Simulation Parameters

Parameter	Value
Simulator	OMNet++
Protocol Studied	IP Multicast, NICE, BinCast
Simulation Time	500 sec
Network	Overlay network
Bandwidth	Variable
Traffic type	CBR (TCP)
Data payload	Bytes/packet

A. Network Setup

In Fig. 5, each poset consist of seven children nodes and one poset head node. Nodes are arranged by poset value, it is calculated by delay x Bandwidth product. There is a single server, which is used to start the multicast session.

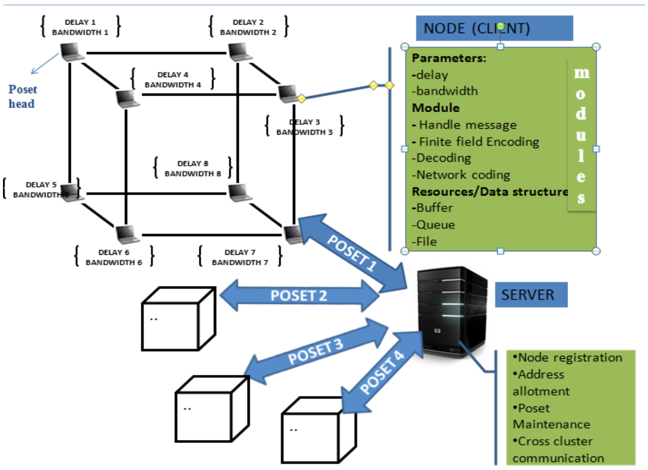

Fig. 5. Network setup

VI. Performance Evaluation

Performance evaluation matrices are delay and bandwidth. *Delay*: On plotting the number of receivers on the *X* axis and End to end delay on the *Y* axis:

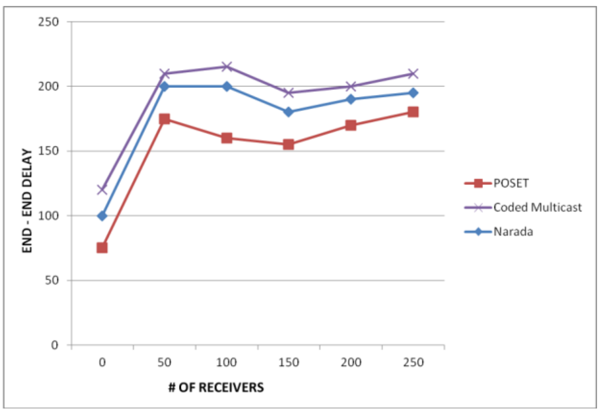

Fig. 6. End-end delay comparison.

In Fig. 6. End-End delay is very high in Coded Multicast protocol because the data distributions are sequential and nodes are arranged by hierarchical clustering technique. Narada is mesh architecture protocol, every node is connected to every other node and hence multiple link transmission and unwanted transmissions occur. But this is better than coded multicast. The proposed POSET architecture is processing two packets at a time, so the delay is obviously decreased compared with both existing protocol.

Throughput: On plotting the packet size on the *X* axis and Bandwidth Utilization on the *Y* axis:

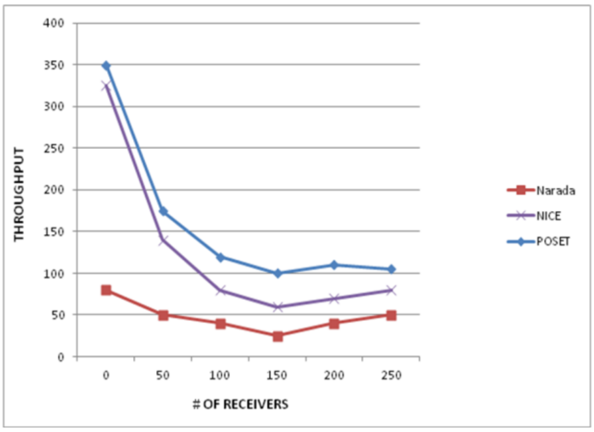

Fig. 7. Throughput comparison.

In Fig. 7, there is no technique used in existing protocols like NICE and Narada, as contradicting to the traditional methods, network coding combines two or more packets and transmits, hence promoting maximum bandwidth utilization. For the same above reasons, POSET architecture provides a higher throughput.

VII. Conclusion

Traditional protocol is used only tree and mesh kind of topology structure for data distribution. In this paper, mesh-poset topology structure has been proposed. In a poset Nodes are arranged by the lattices properties. Poset value is calculated by delay bandwidth product. This mesh-poset topology embedded with network coding technique which is used reduced delay and increases throughput. We have compared and showed our protocol is better than existing protocols.

References

[1] S. Deering, "Multicast routing in a datagram internetwork," Standford University, Standford, CA, USA, 1992.

[2] J. Moy, *Multicast Extension to OSPF*, 1994.

[3] D. Waitzman and C. Partridge, "Distance vector multicast routing protocol," Stanford University, Nov 1988.

[4] Internet group management protocol, *version 2* (*IGMPv2*), 1996.

[5] S. Deering *et al.*, "Protocol independent multicast-sparse mode (PIM-SM): Motivation and architecture," University of Southern California, 1998.

[6] J. Nicholas, A. Adams, and W. Siadak, "Protocol independent multicast–dense mode (PIM-DM): Protocol specification," 2004.

[7] M. Hosseini, D. T. Ahmed, S. Shirmohammadi, and N. D. Georganas, "A survey of application-layer multicast protocols," *IEEE Communications Surveys*, 3rd Quarter, vol. 9, no. 3, 2007.

[8] Takahiro, Taku, and Tetsuya, "Survey of network coding and its applications," *IEICE Trans Commun*, vol. E94-B, no. 3, pp. 698-717, March 2011.

[9] Y. Zhu, B. C. Li, and J. Guo, "Multicast with network coding in application layer overlay networks," *IEEE Journal on Selected Areain Communications*, vol. 22, no. 1, Jan 2004.

[10] W. Li, K. Hao, and B. B. Wang, "An algorithm of application layer multicast base on linear network coding," *IEEE Consumer Electronics, Communications and Networks*, 2012.

[11] Q. M. Li, D-M. Chiu, and J. C. S. Lui, "On the practical and security issues of batch content distribution via network coding," in *Proc. 14th IEEE International Conference on Network Protocols*, Nov. 2006, pp. 158-167.

[12] N. Cai and R. W.Yeung, "Secure Network coding," in *Proc. IEEE International Symposium on Information Theory*, 2002.

[13] M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, no. 10, October 2006.

[14] T. Chan and A. Grant, "Capacity bounds for secure network coding," in *Proc. IEEE Australian Communications Theory Workshop*, 2008, pp. 95-100.

[15] C. Fragouli1 and E. Soljanin, "Network coding applications," *Foundations and TrendsR in Networking*, vol. 2, 2007, no. 2.

T. Ruso received the B.Sc degree from Department of Computer Science, Aditanar College of Arts and Science, Tiruchendur, Tamilnadu, India, in 2002, the M.Sc degree from Department of Computer Science, Annamalai University, Tamilnadu, India, in 2005, the M.E degree from Department of Computer Science and Engineering, Crescent Engineering College, Chennai, India, in 2007. He is currently pursuing Ph.D degree in Anna University, Chennai, India. His research interests include Application Layer Overlay Multicast, Multiparty Video Conference and Adhoc Network.

C. Chellappan received his Ph.D in the area of Database Systems from Anna University, Chennai, Tamil Nadu, India where he is currently the Professor of Computer Science and Engineering Department. His research interest includes Mobile Computing, Sensor Networks and Parallel system scheduling. He is also a Principal Investigator of project(Rs One Crore) on "Basic directed Collaborative Research in Smart and Secure Environment" sponsored by National Technical Research Organization, New Delhi, Govt of India, 2007-2011.