International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 4, April - 2013

Use Of Random Network Coding Inpeer-To-Peer Networks – Review

Ms. Manjiri U. Karande
VBKCOE, Malkapur

Mr. Bharat
K.ChaudhariVBKCOE,Malkapur

Mr. Mahesh V. Shashtri
VBKCOE,Malkapur

Abstract

Random network coding has seen practical and real-world applications in peer-to-peer (P2P) networks, in which overlap network topologies are formed among participating end hosts, called peers. With random network coding, network nodes between the source and receivers are able to not only relay and replicate data packets, but also code them using randomly generated coding coefficients. There is a new trace collection protocol that allows operators to diagnose peer-to peer networks. The power of progressive encoding to increase block diversity and tolerate extreme block losses by introducing redundancy in the network. The protocol controls the redundancy presented through progressive encoding. The random network coding is beneficial in P2P network.

1. Introduction

Random linear network coding is randomized coding method that maintains "a vector of coefficients for each of the source processes," which is "updated by each coding node". In other words, random linear network coding requires messages being communicated through the network to be accompanied by some degree of extra information. With random network coding[1], it have first considered that random network coding can be applied in real-world networks, by dividing a stream of information into generations, and by performing random linear coding within each generation. Coding coefficients may be carried by packets themselves before transmission. It has been concluded that random network coding is robust to packet loss, delay, as well as variations in the network topology and capacity, and that sessions with random network coding can achieve close to the theoretically optimal performance. With random network coding, have first considered that random network coding can be applied in real-world networks, by dividing a stream of information into generations, and by performing random linear coding within each generation. Coding coefficients may be carried by packets themselves before transmission. It has been concluded that random network coding is robust to
packet loss, delay, as well as variations in the network topology and capacity, and that sessions with random network coding can achieve close to the theoretically optimal performance.

Naturally, an excellent scenario where network coding may be applied in practice is peer-to-peer (P2P) networks. A peer-to-peer (abbreviated to P2P) computer network is one in which each computer in the network can act as a client or server for the other computers in the network, allowing shared access to various resources such as files, peripherals, and sensors without the need for a central server. P2P networks can be set up within the home, a business, or over the Internet. Each network type requires all computers in the network to use the same or a compatible program to connect to each other and access files and other resources found on the other computer. P2P networks can be used for sharing content such as audio, video, data, or anything in digital format.In P2P networks, end hosts (from servers to smartphones),called peers, organize themselves in overlap topologies, in which packet transmission on each of the overlap links is free of errors, it uses the transport protocol, such as transmission control protocol (TCP). Peers are also computing devices that are potentially capable of coding packets by implementing network coding in software, without the need of revising existing switches and routers in the Internet.Peer-to-peer applications have been successfully deployed to provide many tasks such as content distribution, live streaming, distributed computation and collaborations. The main advantages of peer-to-peer architectures are scalability, flexibility to failure and easiness of use. The general way to characterize and diagnose peer-to peer networks is to collect periodic statistics from the peers. For example, UUSee Inc.[9] is a live peer-to-peer streaming provider that depends on sorting servers to collect and aggregate snapshots periodically sent by each peer. Every five or ten minutes, each peer sends a UDP packet to the server containing vital statistics. However, the server bandwidth is not sufficient to handle such excessive amount of data. UUSee trace collection completely shuts down when it cannot handle the load of periodic snapshots. Certainly, it is not a scalable trace collection protocol. The efficiency of a trace collection protocol depends on the accuracy of the aggregated snapshots which is defined by the completeness of the measurements [3]. The goal is to collect snapshots from all the peers even those who have left the session before the time of collection. In other words, an efficient trace collection protocol should be able to capture the dynamics of the peers which is a critical parameter for operators that allow monitoring of network performance.Network coding has been proposed to broadcast or spread or circulate the traces in order to increase data diversity and to be hard to losses. It is the challenge to utilize network coding in a way that allows the protocol to scale and at the same time, to increase the diversity of the exchanged blocks.

In this paper, a new trace collection protocolthat uses random linear network coding to exchange and store the snapshots in the network. This protocol allows continuous trace generation and arbitrary trace propagation by the peers. The peers distribute coded snapshots and collect or store them in a decentralized fashion in the peer-to-peer network. The server periodically analyses the peers using a small fixed bandwidth in order to reconstruct the collected snapshots. The peers cooperate in this process by allocating cache capacity to store snapshots generated by other peers. The principle of network coding is a model shift to allow coding at network nodes between the sources and receivers in a communication session [2].

2. What is network coding?

Network Coding can be defined as the coding at the node present in the network. There are various definitions of network coding. By coding at these nodes, it means random, fundamental mapping from inputs to outputs. Another feasible definition of network coding, is coding at anode in a network with error-free links.Most of the work in network coding has determined around a particularform of network coding: random linear network coding. Randomlinear network coding is randomizedcoding method that maintains “a vector of coefficients for each of thesource processes,” which is “updated by each coding node”. In otherwords, random linear network coding requires messages being communicatedthrough the network to be attended by some degree of extrainformation.Another definition of network coding, is coding at a node in apacket network where data is divided into packets and network codingis applied to the contents of packets. Generally, it is coding abovethe physical layer.

2.1. Is network coding beneficial?

Obviously the Network coding is beneficial for transferring information. The network coding can improve throughput, robustness,complexity, and security.The network coding can be explained by using butterfly network as shown in figure 1

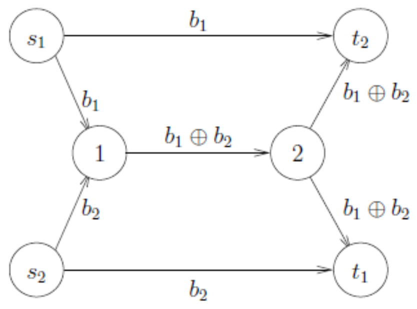

Figure 1. The modified butterfly network

In this network, every edge represents a directed link that is capable of carrying a single packet. There are two packets, $b_1$and b_2, present at the source node $s_1$ands_2, and we want to communicate the contents of these two packets to both of the sink nodes, $t_1$and t_2. Both sink nodes wish to know, in full, the message at the source node. In this network it is considered that, the desired multicast connection can be established only if one of the intermediate nodes i.e., a node that is neither source nor sink breaks from the traditional routing pattern of packet networks, where intermediate nodes are allowed only to make copies of received packets for output, and performs a coding operation. It takes two received packets, forms a new packet by taking the binary sum, or XOR, of the two packets, and outputs the resulting packet. Thus, if the contents of the two received packets are the vectors $b_1$and b_2, each consist of of bits, then the packet that is output is $b_1 \oplus b_2$, formed fromthe bitwise XOR of $b_1$and b_2. The sinks decode by performing furthercoding operations on the packets that they each receive. Sink $t_1$recoversb_2 by taking the XOR of $b_1$and $b_1 \oplus b_2$, and likewise sink $t_2$recovers $b_1$by taking the XOR of b_2 and $b_1 \oplus b_2$.

3. Random network coding in P2P content distribution

One of the best examples of random network coding in P2P bulk distribution is BitTorrent. The BitTorrent provides the large data file to the users over the network. So in the network to make easy to send the data, the large data file break into small pieces or blocks. All these blocks are then transfer to the peers randomly and at the end all data is collected linearly. In this, it allow peers to work together so that large files can be distributed from one peer to a large number of promising receiver, without the help of dedicating server. In P2P bulk distribution a simple method is used that is large file is divided into blocks and peers are connected in random mesh topology. Now again take the example of random gossiping, as its name suggests, in gossiping each peer transmit a subset of the blocks obtained from its neighbour that are selected randomly [1].

In random topology the P2P network form by end hosts who connect with one another using overlay links. If anyone want to distribute a bulk content such as large file, from one peer to another subscribing peers in a topology. So consider a network having n peers who wish to receive a large file. This file is split into or divided into k blocks. Each peer uploads to neighbour that are selected randomly. Let us assume that the time is measured into rounds. The question may arises, how many rounds are needed for all peers to receive a copy of file?

The same philosophy is used in random gossiping which considers the problem of n people spreading a rumor initially held by only one person. How many rounds are needed for everyone to receive the rumor? It is found that if each person who has already received the rumor communicates it to a person chosen at random and independently of all other past and present choices, it takes $\log_2 n + \log n + O(1)$ rounds for the rumor to reach all n persons. Such a random target selection protocol is also called the random phone call. The main difference between that two that is in random gossiping and P2P content distribution is that In P2P content distribution multiple blocks are need to be distributed while in random gossiping a single rumour is distributed. This is the main difference between that two [1].

In random gossiping, in each round, each peer communicates with another target peer which is selected uniformly at random from the whole network. Each peer upload at most one of the blocks it possesses. Within such a model, at least $k + \log_2 n$ rounds are needed to circulate all k blocks from a single source to all n peers [1]. The perception is that it takes at least k rounds for the source to issue the last block, and further $\log_2 n$rounds for that block to reach all n peers.

3.1. Using Random Network Coding

The Random network coding reduces the number of round needed for distribution of k blocks in a random gossiping. It is found that if peers can linearly combine all the blocks it has already received so far using randomly generated coding coefficients, and then transmit such coded blocks to other peers, the amount of time required to distribute a large file to all the peers in the network may be reduced. A simple example of distributing two blocks is given in Figure. 2.

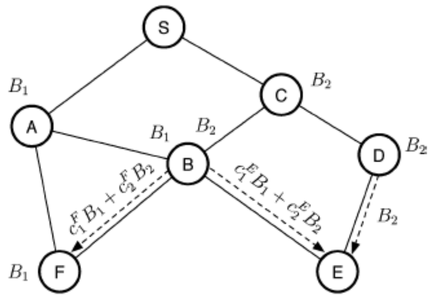

Figure 2. An example of distributing two blocks B1 and B2 with network coding, S is source peer

Suppose that peer B has received blocks B_1 and B_2, and peer D has received block B_2. Although both peers B and D can serve peer E at this point, they may end up transmitting the same block B_2 to peer E, since there is no connection between peers B and D, and their transmission can hardly be coordinated. In this case, the upload bandwidth of peer B is wasted. With the use of network coding, however, peer B can transmit to peer E a coded block $c_1^E B_1 + c_2^E B_2$, with coefficients c_1^E and c_2^E randomly chosen. Peer E can solve for B_1 using the received blocks B_2 and $c_1^E B_1 + c_2^E B_2$.So without coding the peer B may transmit B_1 to peer Fwhich is already possess by F. Hence peer B uses random network coding and transmit to peer F another randomly encoded block $c_1^F B_1 + c_2^F B_2$ which is always beneficial to peer F if c_1^F and c_2^F are appropriately selected [1]. Naturally, the use of random network coding has increased the diversity of blocks being transmitted.

In random network coding for the simplicity the block in a file is divided into multiple generations and network coding is only performed within the same generation. To be more specific, the original file with F bytes is divided into G generations, each of which is further divided into m blocks, referred to as the generationsize. There are a total of M = G.moriginal blocks, each with a size of k =F/M bytes[1].

3.2. Network Coding for Random Gossiping

With the use of random network coding, it is intuitive to see that coded blocks are equally useful at a receiver, as the receiver only needs to accumulate a sufficient number of coded blocks, which is easier than collecting a number of distinct original blocks from its neighbors. When original blocks are collected by a receiver, it may occur that a rare block is harder to run into using random gossiping, which leads to a longer time to finish collecting all the blocks in a file. By using the random network coding the receiver only need to hold a bucket until it is full. All the coded blocks are equally useful at a receiver because the receiver has to only accumulate or collect a sufficient number of coded blocks. And it is easy than collecting a number of distinct original blocks from its neighbours. When original blocks are collected by a receiver, it may occur that a rare block is harder to run into using random gossiping, which leads to a longer time to finish collecting all the blocks in a file. Random network coding simplifies the design of block selection protocols, as it enables the “blind” transmission of coded blocks, without the need for any settlement of blocks between a pair of peers. In this the random phone callmodel [5] has also been considered, where each peer chooses a random target from the entire network to upload the blocks. It transmits only one block in a single round. It is assumed that each peer has one of the k blocks initially. If each peer linearly combines the already received blocks using random coefficients and transmits this coded block to its target then the time for all n peers to receive all k blocks isck + $O(\sqrt{k} \log k \log n)$ rounds, where c is a constant. This essentially shows that by blindlytransmitting coded blocks and selecting communication partners randomly without any form of reconciliation between peers, the time to disseminate all k blocks to all the peers is linear in terms of k [1].

4. Trace Collection Protocol

The new protocol is used in network coding called trace collection protocol.In this protocol, the randomized linear network coding is used for traces exchange and storage. Network coding increases the diversity of the data blocks and the tolerance to block losses upon peer departures. Network coding is performed within segments, where each segment is defined by each peer as the set of blocks forming their snapshots. Random linear combinations are applied to each segment, *i.e.* to the blocks generated by the same peer. example, a peer that has received blocks, that belong to the same segment, generates new coded blocks by the linear combinations of the received blocks over a Galois field GF(2^q) [2], using randomly chosen coefficients. It sends those blocks to neighbors and stores one coded block to be sent to the server once investigated. The segment size is a key factor in the design of the trace collection protocol. The trace collection protocol is used in random linear network coding on the snapshots disseminated in the network. It is assumed that all the peers allocate a cache capacity to store snapshots from other peers. The participating peers encode blocks that belong to the same segment, hence generated by the same peer. Theygenerate and distribute snapshots independently without any time interval restrictions [2].

4.1. Data Block Format

The format of coded block data is given in following fig 3. It addresses progressive encoding used in trace collection protocol.

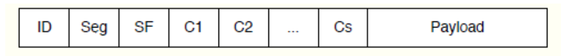

Figure. 3 Coded data block format

The format contains ID which represents the ID of the peer that produced the trace associated with the data block. The “Seg” part indicates the segment number defined by the peer that has produced the block. The “SF” entry represents the spreading factor that defines how far should the blocks associated with this segment propagate. And the coefficients that is C1, C2… Cs used in the encoding process is appendedwith the payload of the data block.The peers only disseminate sufficient data blocks in the network so that the server can decode their traces. On the other hand, the server does not send any acknowledgment to the peers; instead it periodically collects data blocks cached in the network and reconstructs original segments when possible. Otherwise, such messages would lead to significant overhead when used [2].

4.2. Protocol Description

The main goal is to control the redundancy introduced by the dissemination of the traces in the network and to efficiently store the coded blocks in the peers’ caches until they are collected by the server. The spreading factor entry, shown in Figure 2, that corresponds to the group of blocks forming a segment k. Its value is set by the peer that has generated the segment and is modified whenever new blocks are added to the segment. The spreading factor determines the number of blocks that should be disseminated in the network. In other words, it is an indicator of the number of peers that should be reserving an entry in their caches for that segment k. In the trace collection protocol, the spreading factor of a segment depends on its size, defined by the number of blocks it contains, and the neighbours’ departure rate. In fact, coded blocks belonging to a larger segment should be cached more frequently in the network in order for the server to decode them. The higher spreading factorguarantees better tolerance to peer dynamics. Blocks should spread further in the network to resist losses due to high level of peer departures [2]. In this protocol, peers choose the appropriate spreading factor for the newly generated blocks based on the local view of neighbour dynamics. Hence, the number of coded blocks disseminated in the network adapts to the dynamic of the peers.

The spreading factor of a segment is calculated using a logarithmic function as shown in Equation (1). A logarithmic function to take into account previously disseminated blocks of segment k. Those blocks, with different number of coefficients, are used in the decoding process by the server. The spreading factor is defined as

$$SF=\alpha(d)log_2(\beta n_c+1) \quad (1)$$

Where$n_c$is the number of coefficients of segment k. The parameters α(d) and βdetermine the redundancy or the number of coded blocks to be injected in the network. The variable d is the dynamic percentage rate measured using the local view of neighbour dynamics. In our protocol, α(d) is a step function as shown in Equation (2). [2] A peer measures its neighbour dynamics and modifies the spreading factor based on its sensitivity indicated by the percentage levels pl.

$$\alpha(d) = \begin{cases} \alpha_1 & d \leq p_1 \\ \alpha_2 & p_1 < d < p_2 \\ \alpha_l & d > p_l \end{cases} \quad (2)$$

The objective is to effectively reserve the traces in the network and communicate them in a manner that opposes high level of peer dynamics and allows the server to reconstruct the traces by periodically pulling a fixed amount of arbitrarily blocks from the network. To implement the trace collection protocol an encoding method is used which is called progressive encoding method. This method can be explained by algorithm given below:

4.3. Algorithm

Sending Original coded blocks

```
M← Collect measurements
$\hat{B}$ ← divide M and packetize it into block of 1KB
K ← previously disseminated segment or new segment ID
$\hat{B}_k$ ← $\hat{B}_k \cup \hat{B}$
$SF = \alpha(d)\, log_2(\beta \times sizeof(\hat{B}_k) + 1)$
for u = 1 to sizeof(Neighbors)
    $SF_u$ ← Disseminate blocks s.t. $\sum SF_u = SF$
    for i = 1 to $SF_u$
        C ← sizeof($\hat{B}_k$) random coefficients
        b = C × $\hat{B}_k$
        Send b to peer u
    end for
end for
```

Receiving Coded blocks

```
$\hat{B}$ ← received coded blocks
$SF$ ← retrieved message spreading factor
$SF = SF - 1$
ID ← Retrieved message source ID
k ← Retrieved message segment ID
j ← cache entry for segment k from peer ID
$\hat{B}' \leftarrow \hat{B}_j \cup \hat{B}$
for u = 1 to sizeof(Neighbors)
    $SF_u$ ← Disseminate blocks s.t. $\sum SF_u = SF$
    for i = 1 to $SF_u$
        C ← sizeof($\hat{B}'$) random coefficients
        b = C × $\hat{B}'$
        Send b to peer u
    end for
end for
C ← sizeof($\hat{B}'$) random coefficients
if j = = ∅
    if cache is full
        j ← oldest cache entry
    else
        j ← new cache entry
    end if
end if
$\hat{B}_j$ = C × $\hat{B}'$
```

In this, a peer produces new measurements to be collected by the server. As a result, it generates traces and divides them into blocks of size 1 KB each which fits in a single UDP packet. Then, it adds those blocks to its segment **k** that was previously disseminated in the network. After modifying the segment and increasing its size, the peer recalculates the spreading factor ***SF*** using Equation (1). So, when it decides to inject those newly generated blocks in the network, it sends coded blocks from segment **k**. A peer can decide to switch to a new segment when $\boldsymbol{n_c}$ reaches a maximum value that leads to significant coefficient overhead. Note that the blocks exchanged in the network that belong to the same segment can have different number of coefficients. Such blocks are encoded together after appending a corresponding number of zeros to adjust their sizes [2].

Upon receiving blocks from a segment **k**, a peer retrieves the spreading factor from the messages. It then applies random linear combinations to the newly received blocks and the cached blocks that belong to that same segment **k**. According to the retrieved spreading factor, it sends coded blocks to neighbours and sets their appropriate ***SF***entries. Finally, it stores a coded block by replacing previous cache entry of thesegment k, if it exists. Sometimes it may happen that the cache is full, at that time it replaces the oldest segment entry in the cache memory. So it increases the segment size, andthe blocks propagate further in the network replacing previous versions of that same segment.The number of disseminated blocks is determined by thespreading factor set by the peer that has produced the traces. The blocks are increases and the protocol allows the server to decode a large segment of snapshots and limits the number of linear dependent blocks in case of a small segment size by controlling the number of distributed blocks. And hence this protocol decreases the chance of linear dependency when there are few coded blocks andincreases the chance of decoding a segment when it contains a large number of coded blocks. In segment replacement mechanism, progressive encoding solves the problem of data block caching, where the peers have to decide whether to keep an existing block or replace it with a new one. The main principle behind using progressive encoding is to increase blocks multiplicity in order to spread traces to as many peers as possible without introducing many block dependencies at the server. By encoding previously disseminated blocks with newly generatedblocks, blocks variety in live trace generation. On the other hand, the server collects blocks belonging to segments that are increasing in size as it periodically check out the network and deletes them from the peers' caches. With the help of progressive encoding, such cached blocks are more meaningful since they help decoding segments that are increasingly containing newly generated blocks. It also, the problem of decoding all or nothing can be avoided since the server can decode part of the segment if it has collected enough blocks during its periodic inquiring [2].

5. Conclusion

Random network coding is advantageous in P2P network. It is noted that P2P networks may be the most promising application situation for network coding to be applied, only because peers are able to afford the increasedcomputational complexity introduced by network coding.Also random network coding helps in P2P bulk content distribution systems.Network coding increases the diversity of the data blocks and the tolerance to block losses upon peer departures. The trace collection protocol allows continuous trace generation and arbitrary trace dissemination from participating peers. Hence the trace collection protocol helps in coding. In order to be understanding to peer dynamics, the trace collection mechanism disseminates copies of the data generated in the network.

References

[1] BaochunLiandDiNiu "Random Network coding in Peer-to Peer Network: From Theory To Practice" Proceedings of the IEEE | Vol. 99, No. 3, March 2011.

[2] Elias Kchdi, Baochun Li "Incorporating Random Linear Network Coding for Peer to peer Network Diagnosis "NSERC Discovery,CRD and Strategic Grants (RGPIN 238994-06,CRDPJ 379623-08,STPGP 364910-08).

3] C. Wu and B. Li, "Echelon: Peer-to-peer network diagnosis with network coding," in Proc. of IWQoS 2006, New Haven, CT, USA, June 2006.

[4] C. Gkantsidis and P. Rodriguez, Network coding for large scale content distribution,in Proc. IEEE INFOCOM], Mar. 2005, vol. 4, pp. 2235–2245.

[5] S. Deb, M. Medard, and C. Choute, Algebraic gossip: A network coding approach to optimal multiple rumormongering,IEEE Trans. Inf. Theory, vol. 52, no. 6, pp. 2486–2507,Jun. 2006.

[6] P. Sanders, S. Egner, and L. Tolhuizen, Polynomial time algorithm for network information flow,in Proc. 15th ACM Symp. Parallelism Algorithms Architectures, Jun. 2003, pp. 286–294.

[7] T.Ho,R.Koetter,M.Medard,D.Karger,and M.Effros, The benefitsofcoding over routing ina randomized setting,in Proc. Int. Symp. Inf. Theory, 2003,DOI: 10.1109/ISIT.2003.1228459.

[8] P.Chou, Y. Wu, and K. Jain, Practical network coding,in Proc. Allerton Conf. Commun. Control Comput., Oct. 2003. [Online].

[9] "UUSee Inc." http://www.uusee.com/.