www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 2 Issue 9 September 2013 Page No. 2756-2760

Network Coding as an Efficient Technique of Transmitting Data

Khundrakpam Johnson Singh, Usham Sanjota Chanu
Department of Computer Science & Engineering National Institute of Technology, Manipur
Department of Computer Science & Engineering Channabasaveshwara Institute of Technology, Tumkur

Abstract: Network coding is a phenomenon which allows each node in a network to perform some kind of computation. The data sent on a node's output link can be some function or combination of data that arrived earlier on the node's input links. In another, network coding is the transmission, mixing, and remixing of messages arriving at nodes inside the network, such that the transmitted data can be unmixed at their final destinations or at sink. In this paper we take multiple sources which have the capacity of transmitting the data to the next intermediate node. The process of mixing operation is carried out at the intermediate node or the router. The paper also reduces the frequency of mixing the data from different source node by maintaining a fixed random function. At last the sink decodes the mixed data from different sources and reassembles to form the original data.

Keywords: network, encode, decode, router, sink, modulus, intermediate node, router packet.

1. INTRODUCTION

Network coding is a technique which can be used to improve a network's throughput, efficiency and scalability, as well as resilience to attacks and eavesdropping. There are a various method of network coding namely linear network coding [1], random network coding [2] which work effectively in mixing and remixing of the incoming data at the intermediate node and the sink node respectively. In linear network coding butterfly network is often used to illustrate how linear network coding can outperform routing. In Fig.1.two source nodes (at the top of the picture) have information A and B that must be transmitted to the two destination nodes (at the bottom), which each want to know both A and B. Each edge can carry only a single value (we can think of an edge transmitting a bit in each time slot). If only routing were allowed, then the central link would be only able to carry A or B, but not both. Suppose we send A through the centre; then the left destination would receive A twice and not know B at all. Sending B poses a similar problem for the right destination. We say that routing is insufficient because

no routing scheme can transmit both A and B simultaneously to both destinations.

Using a simple code, as shown in Fig.1, A and B can be transmitted to both destinations simultaneously by sending the sum of the symbols through the centre – in other words, we encode A and B using the formula "A+B". The left destination receives A and A + B, and can calculate B by subtracting the two values. Similarly, the right destination will receive B and A + B, and will also be able to determine both A and B.

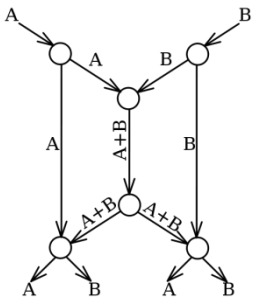

Fig1: Butterfly Network

Random network coding is a simple yet powerful encoding scheme, which in broadcast transmission schemes allows close to optimal throughput using a decentralized algorithm. Nodes transmit random linear combinations of the packets they receive, with coefficients chosen from a Galois field. If the field size is sufficiently large, the probability that the receiver(s) will obtain linearly independent combinations (and therefore obtain innovative information) approaches 1. It should however be noted that, although random network coding has excellent throughput performance, if a receiver obtains an insufficient number of packets, it is extremely unlikely that they can recover any of the original packets. This can be addressed by sending additional random linear combinations until the receiver obtains the appropriate number of packets.

2. NETWORK CODING BENEFITS

Equipped with a definition, we now proceed to discuss the utility of network coding. Network coding can improve throughput, robustness, complexity, and security. We discuss each of these performance factors in turn.

2.1. Throughput

The most well-known utility of network coding—and the easiest to illustrate— is increase of throughput. This throughput benefit is achieved by using packet transmissions more efficiently, i.e., by communicating more information with fewer packet transmissions. The most famous example of this benefit was given by Ahlswede et al. [4], who considered the problem of multicast in a wire line network. Their example, which is commonly referred to as the butterfly network (see Fig.2), features a multicast from a single source to two sinks, or destinations. Both sinks wish to know, in full, the message at the source node. In the capacitated network that they consider, 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 paradigm 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 b1 and b2, each comprised of bits, then the packet that is output is b1 ⊕ b2, formed from the bitwise XOR of b1 and b2. The sinks decode by performing further coding operations on the packets that they each receive. Sink t1 recovers b2 by taking the XOR of b1 and b1 ⊕ b2, and likewise sink t2 recovers b1 by taking the XOR of b2 and b1 ⊕ b2. Under routing, we could communicate, for example, b1 and b2 to t1, but we would then only be able to communicate one of b1 or b2 to t2.

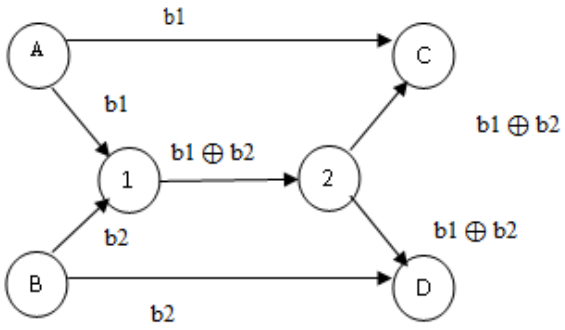

Fig.2. The modified butterfly network

In this network, every arc represents a directed link that is capable of carrying a single packet reliably. There is one packet b1 present at source node s1 that we wish to communicate to sink node t1 and one packet b2 present at source node s2 that we wish to communicate to sink node t2.

2.2. Robustness

2.2.1. Robustness to packet losses

But before we proceed, we address an important issue in packet networks, particularly wireless packet networks, that we have thus far neglected: packet loss. Packet loss arises for various reasons in networks, which include buffer overflow, link outage, and collision. There are a
number of ways to deal with such losses. Perhaps the most straightforward, which is the mechanism used by the transmission control protocol (tcp), is to set up a system of acknowledgments, where packets received
by the sink are acknowledged by a message sent back to the source and, if

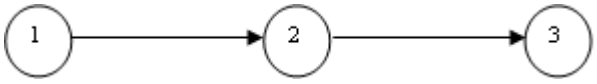

Fig.3. Two-link tandem network. Nodes 1 and 2 are each capable of injecting a single packet per unit time on their respective outgoing links.

the source does not receive the acknowledgment for a particular packet, it retransmits the packet. An alternative method that is sometimes used is channel coding or, more specifically, erasure coding. An erasure code, applied by the source node, introduces a degree of redundancy to the packets so that the message can be recovered even if only a subset of the packets sent by the source is received by the sink.

2.2.2. Robustness to link failures

Besides robustness against random packet losses, network coding is also useful for protection from non-ergodic link failures. Live path protection, where a primary and a backup flow are transmitted for each connection, allows very fast recovery from link failures, since rerouting is not required. By allowing sharing of network resources among different flows, network coding can improve resource usage. For a single multicast session, there exists, for any set of failure patterns from which recovery is possible with arbitrary rerouting, a static network coding solution that allows recovery from any failure pattern in the set.

2.3. Security

From a security standpoint, network coding can offer both benefits and drawbacks. Consider again the butterfly network (Figure 1.1). Suppose an adversary manages to obtain only the packet b1⊕b2. With the packet b1⊕b2 alone, the adversary cannot obtain either b1 or b2; thus we have a possible mechanism for secure communication. In this instance, network coding offers a security benefit.
Alternatively, suppose that node 3 is a malicious node that does not send out b1⊕b2, but rather a packet masquerading as b1⊕b2. Because packets are coded rather than routed, such tampering of packets is more difficult to detect. In this instance, network coding results in a potential security drawback.

3. BACKGROUND

The proposed method works on the simulation of at least four source nodes, two intermediate nodes and a sink node as shown in Fig.4 (a). This example network can be extended to even large networks having multiple source nodes and a sink node as shown in Fig.4 (b). In the proposed system the message from the source are read in multiple of hundred bytes. In case if the remaining message cannot be grouped into hundred bytes the available message is prefixed with "0" to form the required hundred bytes. These organized messages are referred to as the "Type 0" messages.

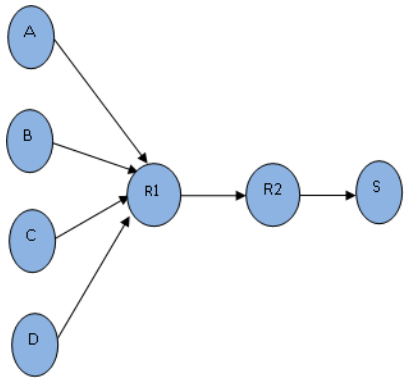

Fig.4 (a).An example network

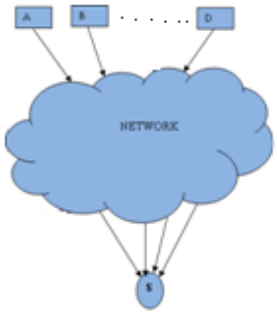

Fig.4 (b). Network with multiple sources and a sink

In the proposed system four packets are required for random coding. The incoming packets are stored in the message queue, which is maintained in the router. The size of the message queue is calculated to check whether it is equal to four or not. If the queue size is four then the random coding continues otherwise a stipulated time period of 1sec is assign as waiting time for the incoming packets. If the waiting time expires then required numbers of remaining packets for random coding are generated as dummy packets. The required numbers of dummy packets are generated by first checking the size of the message queue. Thus the required numbers of dummy packets are generated as in equation 1.

Dummy Packets= 4-sizeof (message queue). (1)

When the required numbers of packets are available in the message queue, the random coding continues. The incoming message to the intermediate node's message queue is referred to as "Type 1" message. Each packet of hundred bytes is taken as one block and is divided into a multiple of twenty five bytes within the block. Each block and sub block is name from A to D and 0 to 3 respectively. Each source node sending the message to the router or intermediate node maintained a unique node ID. The number of blocks obtained at the router from the original message is taken in multiple of the node ID considered. Inside the block the first twenty five bytes from 0 to 25 bytes is taken number 0, the second twenty five bytes from 25 to 50 bytes is taken as number 1, the third twenty five bytes from 50 to 75 bytes is numbered as 2 while the third twenty five bytes from 75 to 100 bytes is taken 3.

After the division of the four blocks in to a multiple of twenty five small sub blocks the composite packet is formed by choosing each sub blocks from four different blocks randomly. The composite packet which is a random combination of different sub blocks has the total size of hundred bytes. The composite packet is again divided into a multiple of twenty five bytes and each sub blocks of the composite packet is given the number from 0 to 3 as above.

The size of each block is taken as 4 (0 to 3), so the representation A:3 means that the third sub block of block A is under consideration. At first the size of each block is calculated and the first decrement of the block size is assign to block A, B, C and D. In order to fill the composite packet in random order of these original blocks (A,B,C,D) a random number is generated starting from 0 to 3 using the math. random function and the sub block in the composite packet are selected randomly from the different blocks.

The randomly generation of number from 0 to 3 makes the arrangement of the sub blocks from the original blocks in the composite packets. The order of the arrangement of the sub blocks of A, B, C and D inside the composite packet as well as the order of selection of blocks for forming the composite packets are computed. Then the contents of different sub blocks from different blocks are shifted to the respective sub blocks of the composite packets according to the above arrangements. The type of message form in the router or intermediate node is referred to as "Type 2" message.

The order of the packets and the order of the blocks are sent to the destination node (sink). The order of the packets and order of the blocks are sent to the destination (sink) so that the sink can re-arrange the information from the composite packets to form the original data. This scheme is not restricted to only four source nodes or to a particular data size, these could be customized.

4. COMPOSITE PACKETS

The composite packets are formed at the intermediate node. It is the mixture of the incoming messages in a particular pattern. In case of multiple source nodes sending the message to the single intermediate node, the composite packet is sent out as a mixture of messages from different sources. In this case the message queue in the intermediate node will take hundred bytes from four sources so that the required four packets are obtained for random coding. After the completion of the formation of the composite packets from the first four source nodes the remaining source nodes are taken. The generation of dummy packets is needed when the router's message queue does not get enough four packets for the

random coding. After the successful formation of the order of the blocks and the order of the packets are sent to the destination along with the composite packets so obtained.

In multi-hop network the information is transmitted from source node to destination node through a number of intermediate. When the composite packets are transmitted from the router to the sink it may face a number of intermediate nodes, in these case the intermediate node need not perform further encoding of these composite packets. The intermediate node will first check whether the incoming messages are encoded or not. If these messages are encoded the intermediate node will just by pass the composite packet to the next hop.

In the proposed system "Type 2" messages are the composite packets, so it does not need to be further encoded. The intermediate node will check the incoming messages for its type. If the message type is "Type 2" then these composite packets are forwarded to the next hop without any further encoding and encryption.

5. PERFORMANCE

The performance of the project is calculated in terms of computational overhead and network overhead. Computational overhead is defined as the extra time taken to meet the particular goal. Whereas the network overhead is defined as the extra memory taken that are required to attain the particular goal. The performance of the proposed system is compared with the without coding or encryption scheme to distinguish the difference between them.

In order to calculate the performance, a simulation of the proposed method is run. The simulation consists of four source nodes which are sending messages to the router. The router will perform the random coding along with the homomorphic encryption of the GEVs. These composite packets along with the encrypted GEVs are sent to the second router. The second router will just by pass the composite packets to the sink. The sink waits to receive four such composite packets, after receiving four composite packets the sink will decrypts the encrypted GEVs. The decrypted GEVs consist of order of the packets and the order of the blocks. The sink uses this information to arrange the sub blocks of the composite packets in an order.

To compute the computational overhead one must know the total time taken to execute the process of random coding and the homomorphic encryption along with time taken for decrypting the GEVs, decoding the composite packets and arranging the messages in proper formats as that of the original messages.

The calculation of the computational overhead requires a start time and the end time of the simulation in terms of milli-seconds. A graph is plotted to find the computational overhead, the x axis of the graph represents the number of packets that are considered, while the y axis represents time taken in milli seconds. In the simulation the total time taken for the hundred packets are computed. As the x axis represents the number of packets taken in the simulation, the total hundred packets are distributed along the x axis with 10 packets forming one unit. The total time taken is arranged along the y axis with 2500 milli seconds forming one unit. The corresponding graph is plotted taking the co-ordinates of the x axis and the y axis. After the plotting of every co-ordinate a line is drawn for every two consecutive points. Thus a computational overhead graph is obtained as shown in Fig.5 (a) and the network overhead is shown in Fig.5 (b).

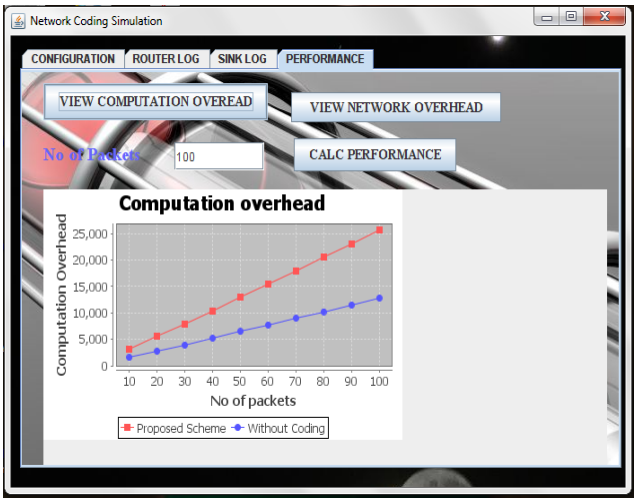

Fig.5 (a) Computational overhead

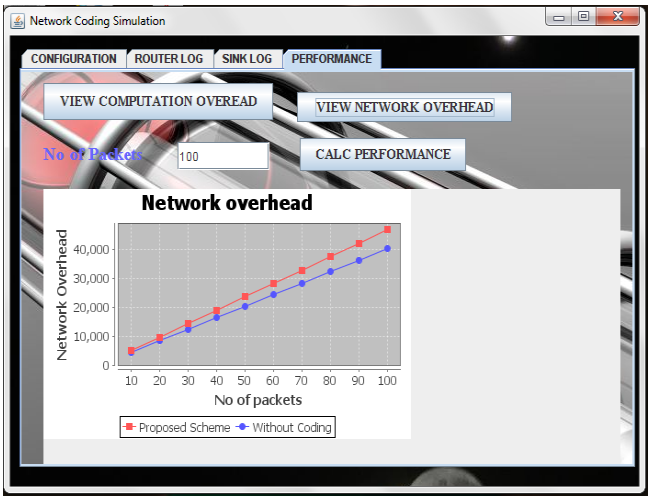

Fig.5 (b) Network overhead

6. Conclusion

The proposed Random Coding and the Homomorphic Encryption scheme shows the formation of the composite packets, which is the random mixing of the incoming messages; the order of the packets and order of the blocks, are encrypted using the homomorphic encryption. In the simulation the simultaneous sending of the messages from different sources is achieved. This makes every instance of composite packet obtained to be the mixture of the messages from different sources. In real time the mixing of the messages is done from a single source as the messages cannot be sent simultaneously from all the sources. The delay in dividing the messages into four blocks is removed by the generation of dummy packets in case of real time. The employment of HEFs on the GEVs in both simulation and real time makes the adversary hard to analyze the order of the formation of the composite packets and thus unable to trace the source by analyzing the message content.

In the future, work can be taken up to improve the performance of the proposed system by limiting the time taken for the transmission. The project can be extended not only text messages, but also the pictures or maps as inputs.

REFERENCE

[1] Michael G . Reed ,Member ,IEEE ,Paul F . Syver-son and David M .Goldschlag " Anonymous Connections and Onion Routing ",IEEE Journal On Selected Areas in Communications .Vol .16 No. 4 May 1998

[2] G. Danezis, R. Dingledine, and N. Mathewson, "Mixminion: design of a type III anonymous remai-ler protocol," in *Proc*. IEEE Symposium on Securi-ty and Privacy, pp. 2-5, May 2003.

[3] X. Wu and N. Li, "Achieving Privacy in Mesh Networks", Proc. of the fourth ACM workshop on Security of Ad hoc and Sensor Networks (SASN'06), pp. 13-22, 2006.

[4] Yanfei Fan, Yixin Jiang, Haojin Zhu, Member, IEEE, Jiming Chen, Member, IEEE, and Xuemin (Sherman) Shen, Fellow, IEEE , "Network Cod-ing Based Privacy Preservation against Traffic Analysis in Multi-Hop Wireless Net-works", IEEE Transactions On Wireless Communi-cations, Vol. 10, No. 3, March 2011

BIBLIOGRAPHY

KHUNDRAKPAM JOHNSON SINGH completed his B.E degree in computer science and engineering from KBNCE in 2010, completed M.Tech from Dayananda Sagar College of Engineering, Bangalore and now working as an Assistant Professor in National Institute of Technology, Manipur in Computer Science and Engineering department. Khundrakpam Johnson Singh is the main author and may be reached at johnkh@gmail.com.

USHAM SANJOTA CHANU completed her B.E degree in computer science and engineering from ICFAI, Tripura in 2010, completed M.Tech from SJBIT, Bangalore and now working as an Assistant Professor in Channabasaveshwara Institute of Technology, Tumkur in Computer Science and Engineering department. Usham Sanjota Chanu may be reached at chanu06atcs012@gmail.com