Journal of Engineering and Applied Sciences 9 (5): 190-197, 2014
ISSN: 1816-949X

A Novel Approach to Incorporate Efficient Security in Content Distribution

[1]K. John Singh, [1]Neha Dilawari and [2]R. Manimegalai
[1]School of Information Technology and Engineering, VIT University, Vellore,
[2]Department of Computer Science and Engineering,
Park College of Engineering and Technology, Coimbatore, Tamil Nadu, India

Abstract: Network coding for content distribution has been in recent, researches lately owing to its wide applications in distribution of shared multimedia content, etc. In network coding, an intermediary node can combine different packets together in order to reduce the number of transmissions and thus increase the overall throughput of the network. Since, attackers tend to inject fake data in order to corrupt the process of content distribution, so that the distribution of information gets hindered and the network resources gets exhausted, content verification becomes a crucial issue when network coding is used. Moreover, it is quite infeasible to use the conventional hash-and-sign methodologies to sign all the data. Therefore, a homomorphic hash function is proposed to overcome the anomaly. However, this technique is quite complex in application and has communication overheads. So, researchers further explore the technique by examining and proposing other methods to aid in reducing computational cost, keeping the security concern in parallel.

Key words: Network coding, content distribution, security, random network coding, epidemic attack, throughput

INTRODUCTION

Network coding is one of the raging subjects in today's information and communication theories. It has gained a lot of attention in the recent researches, since its inception. The core notion of network coding is to allow and encourage mixing of data at intermediate network nodes. A receiver receives these data packets and derives from them, the messages that were originally intended for the data sink. In contrast to conventional ways to function a network that try to avoid collisions of data streams as much as possible, this refined perception implies a richness of surprising results. The incoming packets may be replicated, forwarded and coded due to the ability of coding at the intermediate nodes. This concept is thus, different from the conventional methods where only forwarding of incoming packets was allowed. An important issue in practical large content delivery in a fully distributed environment is how to maintain the integrity of the data in the presence of link failures, packet losses and even malicious attackers. If malicious attackers are able to alter the data while in transmission or interpose arbitrary faked data into the network, it may lead to rapid slow down of the content distribution and may even forbid the users from getting accurate data.

Moreover, network coding has been an emerging topic for content/file distribution in peer to peer networks (Li and Niu, 2011). The basic idea in P2P content distribution protocols is surprisingly simple. Consider a single server distributing a file to a large number of end hosts (peers) over the internet. Instead of uploading the file to every individual peer, the server first divides the file into r data blocks and then distributes these data blocks in an efficient manner by letting participating peers exchange them with one another. The essential advantage of P2P content distribution is to dramatically reduce the file downloading time for each peer. Intuitively, as participating peers contribute their own upload bandwidth to serve one another, the aggregate upload bandwidth in the system is significantly increased, leading to a much faster file distribution process. The study is focused on the efficient security techniques for the content distribution across the network. Many past researches have been done to incorporate network coding for effective content delivery (Gkantsidis and Rodriguez, 2006; Ho *et al*., 2008; Li *et al*., 2003) have further shown that linear coding suffices to achieve the maximum network flow rate. It can be well explained by the popular butterfly network.

The information A and B needs to be transmitted to the 2 destination nodes located at the bottom of Fig. 1. Each of the destination nodes needs to know the complete information presented, as A and B. Also, every edge can carry only one value, either A or B. The overall and complete design is shown in Fig. 1.

Corresponding Author: K. John Singh, School of Information Technology and Engineering, VIT University, Vellore, Tamil Nadu, India

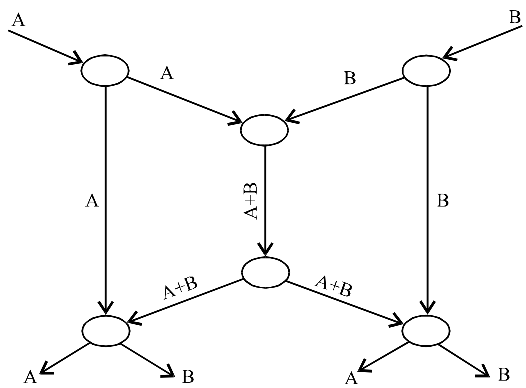

Fig. 1: Butterfly network

As shown, both A and B can be transmitted to the 2 destination nodes simultaneously by encoding them (summing the symbols with a formula as A+B) and transmitting via a centre edge. The destination node on the left receives the original A and encoded A+B and can compute B by performing subtraction on the two values. Similarly, the destination on the right will receive original B and encoded A+B and will also be able to deduce both A and B.

The concept of network coding is young. It was only in the year 2000 that the originative study by Ahlswede *et al*. (2000), marked the birth of network coding. They defined network coding, as an arbitrary function from inputs to outputs. The network model of Ahlswede *et al*. (2000) is a special case of those ordinarily used for informational theory researches and consists of nodes connected to each other via bug-free point-to-point links.

Also in practical content distribution schemes where the network topology is very dynamic that is the nodes may add/withdraw themselves from the network, nodes may fail, etc., random network coding is of true importance (Ho *et al*., 2008).

The main advantage of using network coding for distributing large files (Gkantsidis and Rodriguez, 2005) is that the scheduling of the propagation in the network is much easier. The global information is necessary to determine the right data block to be transmitted to the next node, such that the transmitted packet is useful to the receiver.

With network coding, each node of the distribution network is able to generate and transmit encoded blocks of information. The process of coding introduces the randomization which in turn eases the process of block scheduling during propagation, making the dispersion more effective. This scheme is specifically useful in unstructured overlay networks. In such networks, the nodes use local information to make block forwarding decisions.

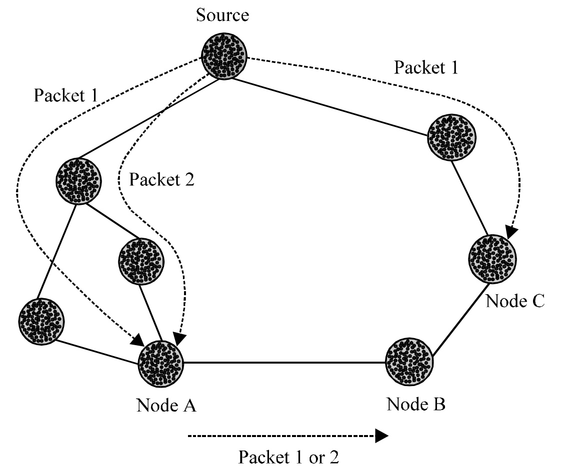

Fig. 2: Utility of network coding (Gkantsidis and Rodriguez, 2006)

Network coding is seen to be a novel approach to improve the throughput of a network topology (Ahlswede *et al*., 2000). The rationale behind network coding is the process of packet encoding by intermediary nodes. Compared to other classical approaches, the approach of network coding optimally uses the network resources and also easy computation in the scheduling scheme may be performed. Lately in many studies, an overview of network coding and a discourse of various internet applications are also suggested.

The following instance is considered to demonstrate the utility of network coding and to show how it enhances the information propagation without a global coordinated scheduler.

In Fig. 2, suppose that node A has received the packets 1 and 2 from the source. Now, if network coding is not employed then there is an equal probability that node B downloads either packet 1 or packet 2 from A. Also, simultaneously when node B downloads a packet from A, node C may independently downloads packet 1. Moreover, the links between nodes B and C cannot be used if node B resolves to recover packet 1 from A and therefore, both nodes B and C will have the same packet 1.

Random network coding has also been into concern lately and much works have been studied on the concept. Random linear network coding preserves an array of coefficients for each of the source processes and is modified by each node. This type of coding requires some additional messages being transmitted over a network along with certain extra information an array of coefficients. Now-a-days packets networks, a type of communication networks, widely being used and can easily accommodate some type of additional information. With the usage of packets, such additional information can be placed in their headers and this strategy has been in general usage these days (e.g., sequence numbers are often placed in packet headers to keep track of order).

Having given the definition, the next step is to put some light on the utility of network coding. Network coding is useful as follows:

Throughput: The first and foremost utility of network coding is increase of throughput. This benefit is attained by transmitting packets more efficiently and effectively. The most famed instance of this utility was given by Ahlswede *et al.* (2000) and is commonly referred to as the butterfly network (Fig. 1). It highlights a packet transmission from a single source to 2 sinks or destinations. Both the destinations desire to know, the full message at the source node. On the other hand, multicast connection on a capacitated network can be created by one of the intermediate nodes whose task is just to create replicas of incoming packets for output. The intermediate nodes may also perform some type of coding by implementing a XOR of the incoming packets to form a new outgoing packet. The destination node then performs decoding operations on the packets that they each receive.

Robustness to packet losses: An important issue next to be addressed is packet loss. Packet loss may happen for different reasons in networks. This includes buffer overflow, link outage and collision. The most fundamental way to handle such packet losses is to set up a system of acknowledgements by the destination nodes to the source nodes upon receiving the packets. Packet loss may occur when the source does not receive any packet acknowledgement from the destination. An alternative method that can be used is known as erasure coding. It is employed by the source node and puts in a degree of redundancy to the packets, so that the message can be recovered even if only subsets of the packets sent by the source are received by the sink.

Robustness to link failures: Network coding is also useful for protection from random link failures. The link failures can be avoided by transmitting a primary and a backup flow for each connection. In this scheme, rerouting is not required. The network resources are shared among various flows and hence network coding becomes useful in optimal resource usage. Static network coding is also a solution to certain link failures.

Complexity: The optimal routing may be considered, as an alternative to network coding but is quite infeasible and complex to obtain. For instance, usage of steiner trees to select minimum-cost sub-graph is more complex than the method employed with network coding that includes low-complexity distributed solutions.

Security: From a security viewpoint, network coding can have the advantages, as well as the disadvantages. Recall the butterfly network again (Fig. 1). Suppose an adversary obtains only the coded packet (XOR of packet 1 and 2). With this coded packet alone, the adversary cannot attain either packet 1 or 2, thus network coding gains an edge over security mechanism. On the other hand, suppose that Node3 is a malicious node that does not transmit the original coded packet but rather a packet disguising, as the coded packet. Now, such fiddling of packets is not detected easily, since the packets are coded and not only routed. Thus, network coding breaches the security concern in a network.

RELATED WORKS

The originative works in network coding brings out the maximum flow in the network if the nodes in a network execute coding on the data they receive from the neighbouring nodes (Ahlswede *et al.*, 2000). This refined outcome has added a new perception in the era of networking, since it has the potential to reach the level of theoretical network capacity via coding, instead of formal method of routing and forwarding.

Further, Ahlswede differs from most conventional multiterminal source coding problems in the following ways:

- No rate-distortion consideration
- Mutually independent sources
- The arbitrary network configuration
- The arbitrary reconstruction requirements

In this study, they have qualified the coding rate region of the single-source problem. The effect is regarded, as the Max-flow Min-cut Theorem for network information flow. Here, the course of discussion is established on alpha-codes. Hence, the outcome can be heightened by taking into concern more universal coding schemes.

Ahlswede *et al*. (2000), demonstrated that a sender node or a transmitter can convey common information to a set of recipient nodes, at a pace attaining the broadcast capacity. On the whole, it is not generally possible to reach this level of communication rate if one permits the interior nodes for routing or forwarding the packets.

Later Li *et al*. (2003), expressed that linear network codes are quite enough to attain the maximum network capacity needed, although the intermediary nodes need not perform linear coding always. In their demonstration, every node that receives information from its upstream nodes, executes some linear combination of the information and passes the solutions to its downstream nodes. But, the major clause in their implementation was that the topology of the nodes in a network must be known beforehand to calculate the accurate linear combinations of network codes. Also, the topology needs to be kept constant throughout the process of content distribution. Furthermore, they followed an algorithm that is exponential to the number of edges in the network.

Koetter and Medard (2001, 2003), also studied the problem of linear network coding. They bettered and enhanced the results by Li *et al*. (2003) and took into concern the problem of failures in the links between the nodes. They discovered that to handle the trouble caused due to link failures, static linear code is suffice if the failure pattern is known beforehand. However, as detected by Jaggi *et al*. (2005), the algorithm proposed by Koetter and Medard (2003) for the construction of code still needs many coefficients to exponentially check a polynomial identity.

The consequences also shows that no overhead of network management is needed for multicast connections but that network management is essentially required for a change of codes which may be necessary in general cases. This type of network management may guide to decide the minimum number of bits required for network management to respond to a failure.

Jaggi *et al*. (2005), proposed the first code construction algorithm that was concentrated around and runs in polynomial time. They also found that there can be huge breaches between the multicast rates obtained with and without coding. Their algorithm comprises of 2 phases. The 1st phase consists of a flow algorithm. It is executed to determine for each sink, a set of edge-disjoint paths from the source to sink. The second phase comprises of a greedy algorithm that visits each edge in turn and designs the linear coding employed for that edge.

They also discovered that the results of Edmonds (1965), depicts that network coding do not improve the attainable transmission rate when all nodes except the source are sinks. They also indicated that if there exists some nodes that are neither the source nor the sinks then multicast can reach upto a rate that is log V times the optimum rate without coding where V is the number of nodes in the network.

It is also depicted by Jaggi *et al*. (2005) that their methodology of code construction can deal with link failures, given that the failure pattern is known well before. In their researches, they dealt with the anomaly of creating robust network codes. They regarded a model much similar to that of Koetter and Medard (2001, 2003).

Another coding named as, random network coding was suggested by Ho *et al*. (2003) in a scenario where the nodes have no idea about the constantly changing network topology and was used as a method to assure the dependability of the network. In their adjusting, random coding was performed by each node and the chances of fortunate recovery at the sinks can be much expected.

The nodes channelize the linear combination incoming information on every outgoing connection defined by randomly chosen coefficient from some field. The receivers need only the linear combination of the processes, at the source for decoding the incoming signals. Random network coding can surpass routing in certain settings. For instance, consider the setting where 2 processes are to be sending from source to the receiver on a grid network that does not include any communication or routing between the nodes. The idea is to maximize the chances of a single receiver node to receive 2 distinct messages. T accomplish the task, the receiver node can transmit one incoming message to its outgoing channel while the other message still on the remaining channel, thus maintaining message variety.

Also, their approach was quite different from the conventional approach of minimizing the rate of transmission, at the source and then re-routing upon the addition of new sources. Their new methodology optimally uses the available maximum network capacity being fully flexible to changes in network topology or addition of new sources.

Ho *et al*. (2003), employed random network coding real-world networks. They did so by splitting an information flow into generations and by executing random linear coding within every generation. The packets need to carry the coefficients before transmission. Consequently, it has been resolved that robustness in packet loss, changes in network topology and capacity, delay can be attained via random network coding with optimum execution. They devised a method in the form of format of a packet that takes out any centralized knowledge of the network topology or the encoding or decoding functions.

The proposed scheme has many advantages. It is wholly decentralized, receiver nodes can decode with loss in packets or without knowing the locations of link failures or without having the slightest idea about the network topology or the encoding functions or also when the nodes or edges are added or removed in a random manner.

This proposed scheme also uses buffers to contemporize incoming and outgoing packets at each node. Random network coding is used to handle large numbers of packets in the buffer and the format of a packet is, such as that includes global encoding vectors to furnish the receiver nodes with the accurate information of decoding the packets in time-varying circumstances.

Gkantsidis and Rodriguez (2005) aimed at another system for large scale content distribution grounded on random network coding. They depicted by simulation that when P2P networks are used, network coding can be around 30% better than server side coding and about 3 times better than uncoded forwarding.

In their researches, they showed the model for large content distribution. This framework can be employed to distribute chunks of the uncoded file or groups of encoded information (encoding at the source or network coding).

They drafted the introductory procedure of this system, underlining some algorithmic parameters that regard its operation.

They considered the problem of on-the-fly Byzantine fault detection in content distribution (Gkantsidis and Rodriguez, 2006). They observed that the techniques for given by Krohn *et al.* (2004) can be used to protect the integrity of the data without knowing the entire content.

Krohn *et al.* (2004), hash functions known as homomorphic hash functions were first brought in to check chunks on-the-fly where source encoding is performed using rate less code. But, these homomorphic hashes are quite expensive and often needs large cryptographic overheads, tempting the malicious codes to infect the network.

Another simple and effective verification strategy was given by Gkantsidis *et al.* (2006). But, it has less security system employed compared to that in (Krohn *et al.*, 2004) and it also has the limitation clause of the data size needed for verification. The data should be proportional to the size of the content and it needs to be distributed beforehand.

Gkantsidis and Rodriguez (2006), a new cooperative security scheme was proposed for content distribution. In this, the users were involved to distribute content and to protect themselves against malicious users by signaling neighbouring nodes when a malicious chunk is fund. Although, the nodes performs security checks infrequently but if at any point in time any malicious block or code is detected in the system by a node then that node alerts and signals the rest of them. Therefore, this scheme dilutes the computation overhead while removing bad blocks. Such cooperative security scheme shows the loss in the efficiency caused by the attackers is limited to the effort the attackers put to corrupt the communication. Also, it is seen that nodes check and verify only about 5% of the packets and are still able to identify bad blocks. The overuse of alerting signals on the nodes can be reduced by the application of a novel scheme that prevents fake and fraudulent alert messages over the network. The alert messages over the links are rapidly verified against the data stored at each node using random masks and mask based hashes and fake alerts are quickly removed thus preventing nodes from performing unnecessary expensive homomorphic hashing checks. In this study, other possible networking coding attacks are also considered.

Ho *et al.* (2008), demonstrated a robust scheme in multicast distributed settings using random network. It consists of an extension of Byzantine modification detection without the use of cryptographic functions.

In this study, they described a scheme which is specifically useful in ad-hoc multicast settings where end nodes pass information to others. The receiving nodes verify if they receive consistent decoded packets. Since, no cryptographic functions are included, the additional computation is minimal. The only necessity is that receiver nodes get one or more unmodified packets whose contents were unknown to the Byzantine attacker (good packets).

Li and Niu (2011), showed that the most promising platform for network coding is P2P networks. This is because the peers are able to yield the enhanced computational complexity put in by network coding. Also, with a recent study of network coding being successfully implied on an operational on-demand streaming system, the application of network coding in P2P streaming systems seems to be more realistic. It is also feasible to implement network coding on commodity hardware, including servers, desktops and even mobile phones.

Li *et al.* (2012), proposed the fundamental scheme based on the Very Smooth Hash (VSH) function. The principle behind VSH and its versions is that it employs usage of smaller primes as group generators. This scheme, thus enhances the computational efficiency of hash functions that involve many exponentiations. In this study, they gave a homomorphic hash function based on the same principle as of VSH. The homomorphic property is attained by re-ordering the bits in the input message chunks before applying VSH.

Epidemic attack is a severe security problem in network-coding enabled Wireless Mesh Networks (WMNs) (Li and Lui, 2013). An epidemic spreading of marred packets can be easily set up by malicious nodes to deplete network resources. Malicious nodes can inject polluted/bogus packets into the wireless network. If an intermediate node is unaware of receiving a polluted packet, it will continue to perform the packet encoding and then forward the encoded but a corrupted packet to its neighbouring nodes. Hence, these polluted packets behaves like an epidemic spread and are easily disseminated throughout the network, importantly degrading the network resources and execution of legitimate flows. As indicated by Dong *et al*. (2009) more about the launching of pollution attack is emphasized along with its detection and prevention.

CONTENT DISTRIBUTION SECURITY SYSTEM

The proposed system is based upon the detection and identification of epidemic attacks in network-coding enabled wireless mesh networks.

When network coding is applied in WMNs, the source first breaks up the file or message into multiple generations. Each generation is further fragmented into n packets which are usually referred to as native packets. Each packet is further divided into encode words.

Here, a brief background on network-coding enabled WMNs is provided, as well as the mechanism of time based checksum batch verification which is used to determine the existence of polluted packets. Certain detection algorithms are proposed which are based on batch verification to identify pollution attackers, as well as the analytical methodology to quantify the performance measures of the algorithms. In MORE (MAC independent Opportunistic Routing) protocol systems, the source node sends packets in generations and each generation contains n native packets. When the source node is permitted to transmit, it will broadcast coded packets which are the linear combination of the native packets instead of directly broadcasting the native packets.

A MORE header is attached to each coded packet which contains a list of potential forwarders. The source node chooses all its downstream nodes which have a lower ETX distance to the destination as the potential forwarders. The forwarder node verifies itself being in the forwarding list on receiving a packet. The packet innovation is also checked by the forwarders. If yes, it makes a number of transmissions wherein each transmitted packet is also a linear combination of all its received packets in the same generation. An acknowledgement to the source is sent by the destination node on receiving n independent packets, being a signal for the source to transmit next generation.

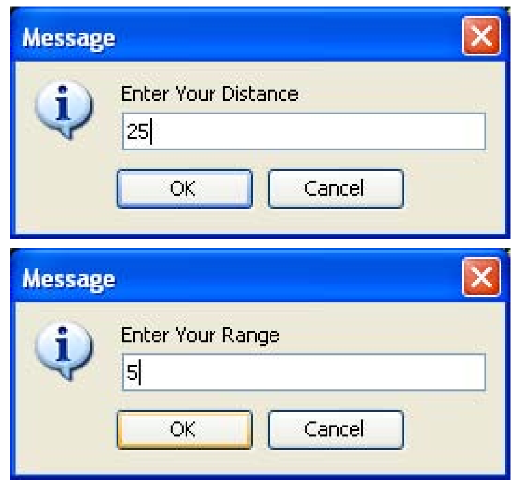

Fig. 3: Determination of neighbouring nodes

A randomized and fully distributed detection mechanism is proposed: Any legitimate node in a WMN can execute our detection algorithms to identify its malicious neighbours.

The malicious nodes are allowed to pretend as legitimate nodes and cooperatively inject polluted packets. The system can be divided into 3 modules:

Network topology: Each node sends hello message to other nodes which allows detecting it. Once a node detects hello message from another node (neighbour), it maintains a contact record to store information about the neighbour. Using multicast socket, all nodes are used to detect the neighbour nodes. The cluster head is elected based on memory, battery and mobility.

The neighbouring nodes are calculated for each node using the distance and range assigned for each node (simulation) (Fig. 3).

Path finding: After choosing the cluster head of a region node establishes their routing protocol to the nearest nodes using the shortest path algorithm. The multiple paths will be found to the cluster head for data transmission.

The source node finds out the entire path to the destination node via several intermediate nodes using shortest path algorithm based on ETX distance. The simulation process done considered Node220, as the source node and the shortest path to the destination node calculated by it is shown in the snapshot (Fig. 4).

Secure data forwarding: During the forwarding phase, a MORE header is attached to each coded packet which contains a list of potential forwarders. The source node chooses all its downstream nodes which have a lower ETX distance to the destination as the potential forwarders. For a forwarder when it receives a packet, it checks whether it is in the forwarder list or not and also checks whether the packet is innovative or not (the packets are encrypted using RSA algorithm). If yes, it makes a number of transmissions wherein each transmitted packet is, also a linear combination of all its received packets in the same generation. For the destination node, if it receives n independent packets, it sends an acknowledgment to inform the source to transmit next generation.

The reason why malicious nodes may imitate legitimate nodes is to thwart the detection, so as to reduce the chance of being detected. On the other hand, for any legitimate node, it strictly follows the routing protocol. Specifically, a legitimate node maintains 2 buffers, verified buffer and unverified buffer. Every time when it is going to forward packets, it only encodes the packets in the verified buffer. On receiving a new packet, it buffers the packet into the unverified buffer. When a checksum packet arrives, it verifies those packets in the unverified buffer based on the time based checksum verification scheme. If the batch verification matches then all verified packets are shifted to verified buffer, otherwise, all packets are discarded. For simulation purpose, HMAC algorithm is being used for checksum verification scheme. Note that by dropping the packets when batch verification does not match, epidemic spreading of polluted packets is avoided so that all packets forwarded by legitimate nodes are valid.

During simulation, Node821 (one of the intermediate nodes in the shortest path to destination node) is pretended to be an attacker node. So, the packets via an attacker node, added to unverified buffer are not transferred to the verified buffer (batch verification fails). This is because the attacker node tends to add some illegitimate data to the received packet and forwards it. Hence, the entire data form source to destination is dropped (Fig. 5).

The detection of an attacker node is then done, as soon as the data is dropped and is isolated from all its neighbours (neighbours are intimidated of the attacker node) and hence is removed from the network topology, thus preventing epidemic attack (Fig. 6).

Fig. 4: Node220-source node

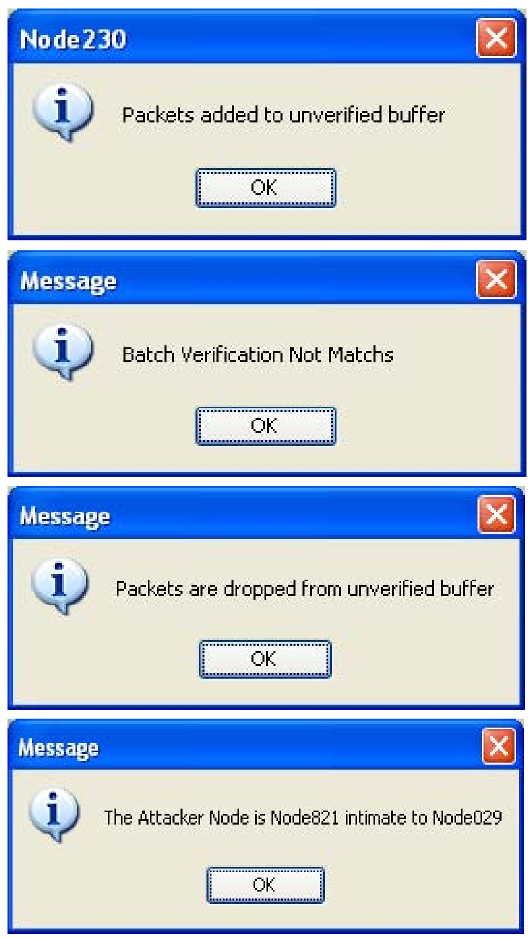

Fig. 5: Detection of attacker

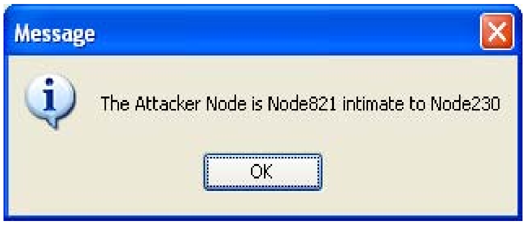

Fig. 6: Removal of attacker node-Node821

CONCLUSION

In network coding enabled WMNs, using this paradigm, high end-to-end throughput can be obtained even if some links along the source-destination path are lossy. However, since multiple nodes which overhear the packet can participate in the packet forwarding, packet collision may occur and thereby reduces the network capacity. Allowing nodes in a WMN to perform network coding opens the door for epidemic (or pollution) attack. Malicious nodes can inject polluted/bogus packets into the wireless network. If an intermediate node is unaware of receiving a polluted packet, it will continue to perform the packet encoding and then forward the encoded but a corrupted packet to its neighbours. Thus, an epidemic attack takes place corrupting the network resources and performance of network flows.

REFERENCES

Ahlswede, R., N. Cai, S.Y.R. Li and R.W. Yeung, 2000. Network information flow. IEEE Trans. Inform. Theory, 46: 1204-1216.

Dong, J., R. Curtmola and C. Nita-Rotaru, 2009. Practical defenses against pollution attacks in intra-flow network coding for wireless mesh networks. Proceedings of the 2nd ACM Conference on Wireless Network Security, March 16-19, 2009, Zurich, Switzerland, pp: 111-122.

Edmonds, J., 1965. Minimum partition of a matroid into independent subsets. J. Res. Nat. Bur. Standards Sect. B, 69: 67-72.

Gkantsidis, C. and P.R. Rodriguez, 2005. Network coding for large scale content distribution. Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies, Volume 4, March 13-17, 2005, Miami, FL., USA., pp: 2235-2245.

Gkantsidis, C. and P. Rodriguez, 2006. Cooperative security for network coding file distribution. Proceedings of IEEE International Conference on Computer Communications, April 23-29, 2006, Barcelona, Spain, pp: 5-5.

Gkantsidis, C., J. Miller and P. Rodriguez, 2006. Anatomy of a P2P content distribution system with network coding. Proceedings of the 5th International workshop on Peer-To-Peer Systems, February 27-28, 2006, Santa Barbara, CA., USA., pp: 1-6.

Ho, T., R. Koetter, M. Medard, D.R. Karger and M. Effros, 2003. The benefits of coding over routing in a randomized setting. Proceedings of the International Symposium on Information Theory, June 29-July 4, 2003, Yokohama, Japan, pp: 442-442.

Ho, T., B. Leong, R. Koetter, M. Medard, M. Effros and D.R. Karger, 2008. Byzantine modification detection in multicast networks with random network coding. IEEE Trans. Inform. Theory, 54: 2798-2803.

Jaggi, S., P. Sanders, P.A. Chou, M. Effros, S. Egner, K. Jain and L.M. Tolhuizen, 2005. Polynomial time algorithms for multicast network code construction. IEEE Trans. Inform. Theor., 51: 1973-1982.

Koetter, R. and M. Medard, 2001. An algebraic approach to network coding. Proceedings of the IEEE International Symposium on Information Theory, Jun 24-29, 2001, Washington, DC., USA.

Koetter, R. and M. Medard, 2003. An algebraic approach to network coding. IEEE/ACM Trans. Networking, 11: 782-796.

Krohn, M.N., M.J. Freedman and D. Mazieres, 2004. On-the-fly verification of rateless erasure codes for efficient content distribution. Proceedings of the IEEE Symposium on Security and Privacy, May 9-12, 2004, Berkeley, CA., USA., pp: 226-240.

Li, S.Y.R., R.W. Yeung and N. Cai, 2003. Linear network coding. IEEE Trans. Inform. Theory, 49: 371-381.

Li, B. and D. Niu, 2011. Random network coding in peer-to-peer networks: From theory to practice. Proc. IEEE, 99: 513-523.

Li, Q., J.C. Lui and D.M. Chiu, 2012. On the security and efficiency of content distribution via network coding. IEEE Trans. Dependable Secure Comput., 9: 211-221.

Li, Y. and J.C.S. Lui, 2013. Epidemic attacks in network-coding-enabled wireless mesh networks: Detection, identification and evaluation. IEEE Trans. Mobile Comput., 12: 2219-2232.