Research on the Network Coding Design and Architecture in General File System

Qin Ma*, Huaiyong Deng

Chongqing Water Resources and Electric Engineering College, Chongqing ,402160, China

Abstract
In this paper, we mainly research on the network coding design and architecture in General File System (GFS). This research work presents a practical network coding approach for the General file system. This approach focuses on network coding and compares it with the replication scheme that General File System uses to provide redundancy. We study the performance in the two cases by evaluating the probability of failure of any chunk, and their ability to recover the original data from any surviving data chunks and we also evaluate the average bandwidth as a number of transmissions required when a request is made by the client to read a data. We observed that with network coding, the system is more robust and resilient to failure and provide better performance than with replication scheme to provide redundancy.
Keywords: NETWORK CODING, DESIGN AND ARCHITECTURE, GENERAL FILE SYSTEM

1. Introduction

The advent of network coding, which is said to "spark networking's next revolution ", has attracted a lot of attention in both the industry and academia. However, recent work has demonstrated the severe harm of attacks such as pollution attacks and entropy attacks. If we cannot solve these problems properly, the communication system that employs network coding may face severe challenges, which could destroy the benefits of network coding, or even result in a worse performance than that in the system with traditional store-and-forward mechanism. In this paper, we focus on the security problems in network coding system. The major contribution of the paper can be summarized as follows: We present a detection scheme to provide authentication for the network coding based directed acyclic network, to resist against pollution attack. First of all, we propose a new construction of the holomorphic message authentication code (MAC), and the security analysis proves that the proposed MAC could achieve the same security with previous methods using a smaller key size. Then, considering the feature of network coding based directed acyclic network, we propose an improved message transmission scheme, which is combined with the proposed holomorphic MAC to detect corrupted packets in the network. Final-ly, the experiments show that, the proposed method indeed has both low computation and communication overhead. We propose a defense scheme to provide authentication for the network coding based dynamic network. Specifically, we focus on how to provide authentication for the network coding based peer-to-peer (P2P) live streaming system, to resist against pollution attacks and entropy attacks, simultaneously. Taking into consideration the high computation efficiency and small communication overheads that are vital requirements for the P2P live streaming, we first propose a holomorphic MAC with smaller key size and lower computation cost, which is called as PMAC [1]. Then, we employ the holomorphic MAC and delayed key dis-closure technique to detect the

corrupted packets in the network, and make the nodes code correctly in accordance with the requirements of randomly linear network coding.

Next, we prove the security of the proposed scheme. At last, the experiments demonstrate the advantage of proposed scheme in reducing the computation and communication cost [2-3]. We propose a key distribution scheme which is suitable for the network coding system. The key idea is that, we make full use of the mixing feature of network coding to distribute keys. Specifically, every communication participant shares a secret with the key generation center (KGC), with which only authorized participant could recover the session key. In addition, the KGC only need to broadcast the messages that are needed for recovering the keys once, even in a public channel. Security analysis demonstrates that the proposed scheme could resist against the insider and outsider attacks, which shows that the proposed scheme achieves both confidentiality and authentication in the transmission of the keys [4].

2. Design Assumption

The computational complexity, bit error rate and achievable rate are analyzed, compared with other network coding schemes. Also the application scenarios of PPNC scheme are discussed. To solve the problem of transmission delay in a multi-user cooperative system, analog network coding is introduced to a multi-user cooperative system, and a cooperative analog network coding scheme is proposed. The cooperation procedure of this scheme is analyzed and the channel capacity of the scheme is derived when the relay node works in amplify-and-forward mode. Compared with traditional cooperation, the influence of the nodes' antenna number on channel capacity is analyzed and the BER performance of cooperative analog network coding is studied under different simulation scenarios. After that, the problem about how to apply physical-layer network coding into multi-user cooperative systems is deep studied. Employing polarization diversity, cooperative quadrature physical-layer network coding scheme is proposed and the cooperation procedure is analyzed. This proposed scheme is a cooperative method based on decode-and-forward and physical-layer network coding. Compared with some other cooperation schemes, the BER and system throughput performances are analyzed under different transmission conditions. Also the effect of different decoding methods on the performance of this proposed scheme is studied. The simulation results reveal that the proposed scheme can effectively improve the system performances. Finally, to further improve the performance of multi-user cooperative systems, the application of PNC in a multi-user cooperative system with multiple-antenna terminals is studied and the cooperative quadrature physical-layer network coding jointed with space-time coding scheme is proposed. The effect of the nodes' antenna number on BER and system throughput performances is analyzed. Also it is compared with traditional decode-and-forward scheme and cooperative network coding scheme. The simulation results reveal that the proposed scheme can effectively improve the system performances, and its performance is the best [5].

The following points present in detail the considerations that were made in the GFS design:

"Components failures are bound to occur all the time. The system is built from many cheap commodity components that frequently fail. Failures of disks, network, power supplies, memory, human errors, etc. are no exceptions given the poor quality and large quantity of the storage machines.

The files are extremely large as compare to the usual standards. It is common to have Multi-GB files and should be managed without difficulty since the system is regularly working with huge data sets with billions of terabytes objects.

The workload generally is made up of two reads: small random reads and large streaming reads.

A large streaming reads typically reads hundreds of KBs, 1MB or more commonly. A small reads usually reads a few KBs at some offset arbitrarily.

Workloads also possess many large, sequential writes that add or append data to files. The sizes of operations are usually the same to those of reads and files are hardly modified once they are written. Small writes in a file are supported at arbitrary positions but can be inefficient.

The system must implement efficiently well-defined semantics for multiple clients that appends to the same file concurrently. With the Introducing atomic append operations multiple clients can at the same time append to a file without synchronizing between them. The files are usually used for many-way merging or as producer-consumer queues. Minimal overhead synchronization of atomicity is vital.

Keeping high bandwidth is more important than low latency. The target applications are so much concern with processing data in bulk at high rate than for an individual read or write. Join designing of the file system API and the applications. The overall system benefits when the applications and file system API are co-designed. The design of the GFS uses merely replication for redundancy and a centralized approach to make the design simple, increase its reliability and gain flexibility.

Modification of GFS files commonly is by appending data although the modifications at arbitrary file offsets are scarce. Thus, unlike to other file systems a large number of files can be considered as being append-only or even immutable (i.e., write once, read many).

The GFS design is optimized for large streaming reads and generally prefers throughput over latency. In case of large working sets, unlike other file systems such as AFS, the GFS doesn't use the client side caching techniques because they are deemed ineffective.

The key idea is that, we make full use of the mixing feature of network coding to distribute keys. Specifically, every communication participant shares a secret with the key generation center (KGC), with which only authorized participant could recover the session key. In addition, the KGC only need to broadcast the messages that are needed for recovering the keys once, even in a public channel.

3. The Algorithm

Network coding allows intermediate nodes in communication networks to en-code messages. The encoding operations at intermediate nodes can provide significant benefits to communication networks, such as increased throughput, reduced network congestion, higher reliability and robustness, lower power consumption, and optimized load balance of network. However, the inherent information-mixing nature of the network coding also makes the network-coding-based applications more susceptible to pollution attacks. In such an attack, an adversary may maliciously forge some messages and inject the polluted messages into the communication network. A small number of polluted messages can cause a large scale of pollution, which makes the sink nodes fail to correctly decode the messages. The pollution attacks can dramatically deplete network resources and significantly decrease network throughput. Moreover, intermediates nodes in the network waste a large number of precious computing resources and bandwidth resources encoding and propagating the polluted messages. There are a great deal of problems pressing for solutions when we apply network coding to practical use. However, one of the most important problems should be the problem of defending against the pollution attacks and other at-tacks in network coding. Imagine if we cannot utilize network coding in a safe environment, then all of the advantages brought by network coding will become meaningless.

Compared to previous work, this new design brings us following primary properties:-It allows both intermediate nodes and recipient nodes to detect and resist the polluted packets that are forged from a normal pollution attack.-It is immediately suitable for the source node to distribute multiple generations using a single public key. Namely, it eliminates the need to redistribute the public keys when the source node begins to distribute a different generation.-It divides the verification process at a recipient node into two steps, which can greatly reduce the sensor nodes' computational overhead under a tag pollution or a repetitive attack.-It does not use any pairing operations in signature generation or verification.-It can be proven secure based on the "lower-level" cryptographic assumptions without random oracles.

The equation of basic function is as equation (1) as follows:

$$\partial_j(C_{ijkl}\partial_k u_l + e_{kij}\partial_k\varphi) - \rho\ddot{u}_i = 0 \tag{1}$$

Under the linear relationship, basic equation is shown in equation (2):

$$\partial_j(e_{ijkl}\partial_k u_l - \eta_{kij}\partial_k\varphi) = 0 \tag{2}$$

The linear differential equation can be expressed into the following simplified forms:

$$L(\nabla,\omega)f(x,\omega) = 0\,,$$
$$L(\nabla,\omega) = T(\nabla) + \omega^2\rho\mathrm{J} \tag{3}$$

In which,

$$T(\nabla) = \begin{Vmatrix} T_{ik}(\nabla) & t_i(\nabla) \\ t_k^T(\nabla) & -\tau(\nabla) \end{Vmatrix},\ \mathrm{J} = \begin{Vmatrix} \delta_{ik} & 0 \\ 0 & 0 \end{Vmatrix},$$

$$f(x,\omega) = \begin{Vmatrix} u_k(x,\omega) \\ \varphi(x,\omega) \end{Vmatrix} \tag{4}$$

$$T_{ik}(\nabla) = \partial_j C_{ijkl}\partial_l\,,\ t_i(\nabla) = \partial_j e_{ijk}\partial_k\,,\ \tau(\nabla) = \partial_i\eta_{ik}\partial_k$$

Consider an infinite situation, we have the equation (5) in the following:

$$L^0 = \begin{Vmatrix} C_{ijkl}^0 & e_{kij}^0 \\ e_{ikl}^{0T} & -\eta_{ik}^0 \end{Vmatrix} \tag{5}$$

Consider the propagation, instead the equation (3) with the following form:

$$C(\mathrm{x}) = C^0 + C^1(\mathrm{x})\,,\ e(\mathrm{x}) = e^0 + e^1(\mathrm{x})\,,$$
$$\eta(\mathrm{x}) = \eta^0 + \eta^1(\mathrm{x})\ ,\ \rho(\mathrm{x}) = \rho_0 + \rho_1(\mathrm{x}) \tag{6}$$

Then we have equation (7) to (11):

$$C^1 = C - C^0\,,\ e^1 = e - e^0\,,$$
$$\eta^1 = \eta - \eta^0\,,\ \rho_1 = \rho - \rho_0 \tag{7}$$

The containing inclusions can be simplified into the following integral equation set:

$$f(x,\omega) = f^0(x,\omega) + \int_V \mathcal{S}(x-x')(\mathrm{L}^1 F(\mathrm{y}') + \rho_1\omega^2 \mathbf{g}(R)\mathrm{T}_1 f(\mathrm{y}')]S(\mathrm{y}')\mathrm{dy}' \quad (8)$$

In view of the following relationship

$$\frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-ik_3 x_3'} dx_3' = \delta(k_3) \quad (9)$$

Equation (8) can be converted into the following form:

$$f(y,\omega) = f^0(y,\omega) + \int_s S(y-y',\omega)\mathrm{L}^1 F(\mathrm{y}',\omega)\mathrm{dy}' + \rho_1\omega^2 \int_s \mathbf{g}(y-y',\omega)\mathrm{J}f(\mathrm{y}',\omega)\mathrm{dy}' \quad (10)$$

In which, S is cylinder cross section, $y = (x_1, x_2)$, and

$$\mathbf{g}(y-y',\omega) = \frac{1}{(2\pi)^2}\int_0^{\infty} \bar{k}\,d\bar{k} \int_0^{2\pi} \mathbf{g}(\bar{k},\omega)\exp(-i\mathrm{k}\cdot(\mathrm{y}-\mathrm{y}'))d\phi \qquad \bar{k} = (k_1, k_2) \quad (11)$$

Suppose $k_3 = 0$, $\mathbf{g}(\bar{k},\omega)$ can be obtained from Equation (8)

For such kind of material, general form of equation (10) is expressed as following equation (12-14):

$$G_{ik}(\bar{k},\omega) = \frac{1}{\rho_0\omega^2}\left[\frac{\beta^2}{\bar{k}^2-\beta^2}\theta_{ik} + \bar{k}_i\bar{k}_k\left(\frac{1}{\bar{k}^2-\alpha^2} - \frac{1}{\bar{k}^2-\beta^2}\right) + m_i m_k \frac{\beta_\perp^2}{\bar{k}^2-\beta_\perp^2}\right] \quad (12)$$

$$\mathcal{g}_{ik}(\bar{k},\omega) = -\frac{1}{\eta_{11}^0}\frac{1}{\bar{k}^2} + \frac{1}{\rho_0\omega^2}\left(\frac{e_{15}^0}{\eta_{11}^0}\right)^2 \frac{\beta_\perp^2}{\bar{k}^2-\beta_\perp^2} \quad (13)$$

$$\gamma_i(\bar{k}_i,\omega) = \frac{1}{\rho_0\omega^2}\left(\frac{e_{15}^0}{\eta_{11}^0}\right)^2 \frac{\beta_\perp^2}{\bar{k}^2-\beta_\perp^2} m_i \quad (14)$$

Our scheme has two main advantages as compared to prior work: It is immediately suitable for the distribution of a large-sized file that consists of multiple generations.-It has a fixed small size public key, and when the length of the file vectors to be sent is changed, it eliminates the need to redistribute the public keys over the network. These significant improvements of previous schemes make our signature scheme more suitable in practice. Finally, we compare our signature scheme with previous signature schemes.

4. The Experiment and Data Analysis

The GFS architecture is based on a master/slave pattern. It is made up of a single master, multiple chunk servers and multiple clients. Figure 1 shows the architecture of GFS.

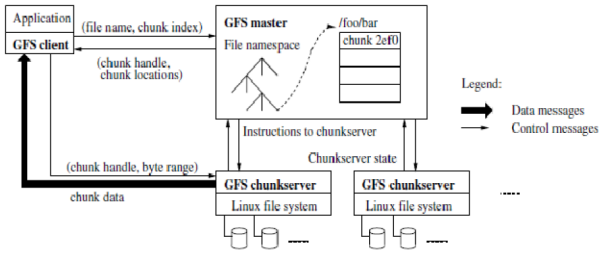

Figure 1. The architecture of GFS

The chunk servers are the slave or workhorses of the GFS. They are in charge for storing the entire chunks on local disks as Linux files. Chunk servers also read or write chunk data which are stated by a chunk handle and byte range. The chunk servers send requested chunk data directly to the client only. In order to ensure reliability of data each chunk is replicated or copied multiple times on multiple chunk servers. By default, only three replicas are stored per chunk server though users can change the setting and choose the number of replicas for different regions of the file namespace [6-8]. The figure 2 shows the writing a chunk process.

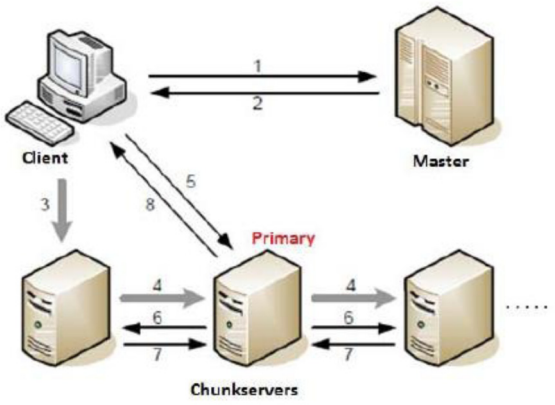

Figure 2. The writing a chunk process

Application begins the write request (filename, data) of the affected file. The clients translates request file from (filename, data) to (filename, chunk index), and sends it to the master. 2) Master responds with chunk handle (or identity) of the primary and other secondary replicas locations. The client caches the data for future mutation and only needs to contact the master once again in case the primary is unreachable or sends a reply it no longer holds a lease. 3) Client sends the data to all the replicas in any order but mostly starting with the closest replica and ending with the furthest. The data is stored in the internal buffer each chunk servers until the data is used or ex-

pire. 4) Write data propagates in a pipelined fashion to the other chunk servers. 5) Once all the replicas in the chunk servers have acknowledged receiving data, the client sends a write request to the primary and which then attributes sequentially serial numbers to all the mutations it receives. The primary administers the mutation to its own local state in a serial order. The mutation order describes the sequence by which the modification is performed. 6) The primary pushes the write request together with the mutation order to all secondary replicas. Each secondary replica administers mutations in the same serial number order done by the primary. 7) The secondary's all reply to the primary replies primary when the modification is successfully completed. 8) The success to the client.

In figure 3, I show a graph that plots the error rate versus the probability when any chunk fails. Error rate here is the probability of the system failure. Therefore the graph above is plotting the probability of system failure versus probability of components failure for both network coding and replication storage schemes.

Conclusions

In this paper, we mainly research on the network coding design and architecture in General File System (GFS). This research work presents a practical network coding approach for the General file system. We propose a key distribution scheme which is suitable for the network coding system. The key idea is that, we make full use of the mixing feature of network coding to distribute keys.

Specifically, every communication participant shares a secret with the key generation center (KGC), with which only authorized participant could recover the session key. In addition, the KGC only need to broadcast the messages that are needed for recovering the keys once, even in a public channel. Our scheme has two main advantages as compared to prior work: It is immediately suitable for the distribution of a large-sized file that consists of multiple generations.-It has a fixed small size public key, and when the length of the file vectors to be sent is changed, it eliminates the need to redistribute the public keys over the network. We study the performance in the two cases by evaluating the probability of failure of any chunk, and their ability to recover the original data from any surviving data chunks and we also evaluate the average bandwidth as a number of transmissions required when a request is made by the client to read a data. We observed that with network coding, the system is more robust and resilient to failure and provide better performance than with replication scheme to provide redundancy.

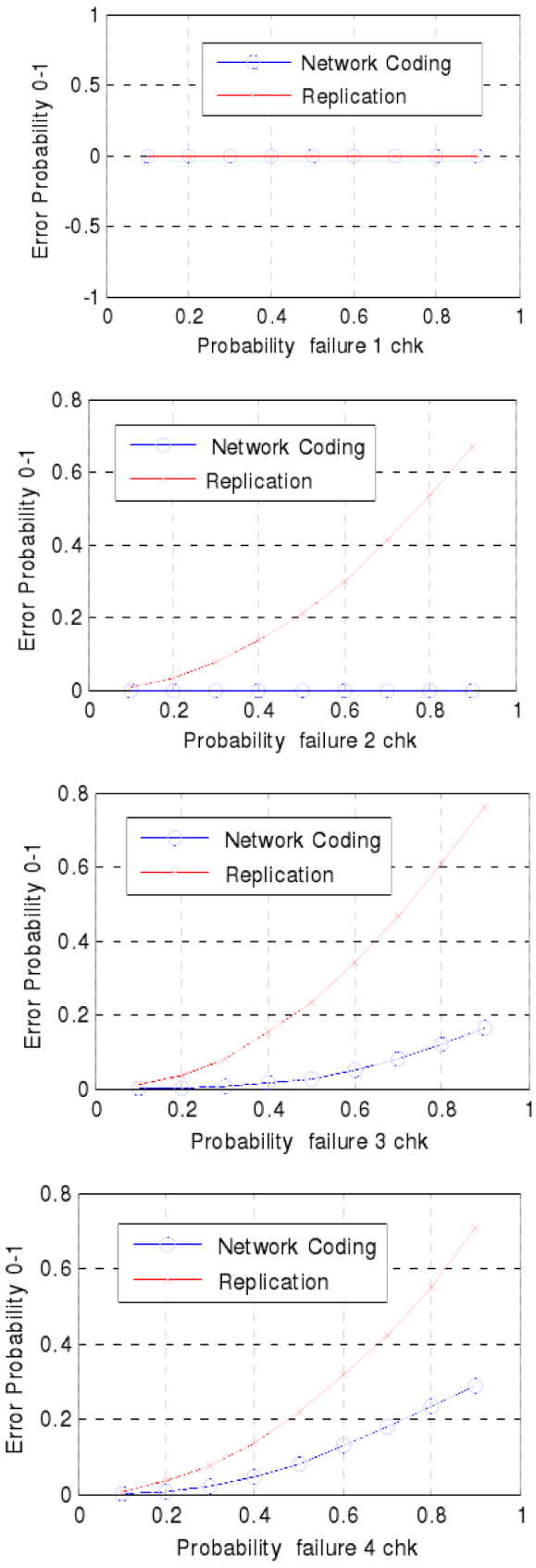

Figure 3. The error rate of the system as a function of the probability of failure of any chunks

Acknowledgements

This work is supported by the the Higher Education reform general project fund of Chongqing, China (No.153283) .

References

1. Shaobin Cai, Zhenguo Gao, DeSen Yang, Nianmin Yao. A network coding based protocol for reliable data transfer in underwater acoustic sensor. Ad Hoc Networks, 2013, pp. 115-128.
2. Zhi-jian QU, Yin-bao XIE, Ju-hui WANG, Xiao-hong LIU. Design of a node architecture for logic-calculation Nased all-optical network coding scheme. The Journal of China Universities of Posts and Telecommunications, 2013, pp. 205-215.
3. Lijun Li, Rentao Gu, Yuefeng Ji, Lin Bai, Zhitong Huang. All-optical OFDM network coding scheme for all-optical virtual private communication in PON . Optical Fiber Technology, 2013, pp. 13-27.
4. Kaikai Chi, Yi-hua Zhu, Xiaohong Jiang, Xianzhong Tian. Practical throughput analysis for two-hop wireless network coding. Computer Networks, 2013, pp. 233-256.
5. Luiz Filipe M. Vieira, Mario Gerla, Archan Misra. Fundamental limits on end-to-end throughput of network coding in multi-rate and multicast wireless networks. Computer Networks, 2013, pp. 5717-5727.
6. Pallavi R. Mane, Sudhakara G. Adiga, M. Sathish Kumar. Performance evaluation of random linear network coding using a Vandermonde matrix . Physical Communication, 2013, pp. 122-132.
7. Yi-jun GUO, Jian-jun HAO, Guang-xin YUE. Reduce the decoding complexity: segment linear network coding. The Journal of China Universities of Posts and Telecommunications, 2013, pp. 206-219.
8. Jian-jun HAO, Yi-jun GUO, Guang-xin YUE, Jian-feng LI. A practical physical network coding scheme over hybrid field. The Journal of China Universities of Posts and Telecommunications, 2013, pp. 20-27.

Research on the Face Recognition in Color Picture Using Characteristic Extraction Based on the Kernel Algorithm

CUI Zhong-Yuan[1], ZHANG Shao-Hui[2]

[1] College of Computer Science and Technology, Zhoukou Normal University, 466001, China
[2]College of Network Engineering, Zhoukou Normal University, 466001, China

Abstract

In this paper, the author researched on the face recognition in color picture using the characteristic extraction based on the kernel algorithm. A variety of methods are proposed in the frame of the sparse representation and dictionary learning to improved algorithms, essentially, which is an extension for the sparse representation to study the