CHAPTER 4

NCMP: NETWORK CODE MINIMIZATION

PROBLEM IN P2P NETWORK

4.1 INTRODUCTION

Several research works in recent years have proved that network coding can also be used to improve the efficiency of content distribution such as application level multicast, P2P streaming applications and reliable large scale multimedia content distribution (Lahbabi et al. 2014; Wang & Li 2007; Wu & Li 2008 and Fragouli et al. 2006).

It is noted that conventional network optimization aims to maximize information flow by utilizing as much link capacity as possible. At the same time network coding begins with the assumption that full link capacity utilization has already been achieved wherever possible and then attempts have been made to further increase the network throughput at the receiver by performing coding at intermediate nodes (Xing & Qu 2013). Although network coding may be performed at all the nodes in some relevant literature, an interesting observation is that a maximum download rate can be achieved by performing the coding operation at a relatively small portion of the network (Xing & Qu 2011). To date, though random network coding emerges as a promising alternative to the traditional methods in theoretical perspective, its benefit in real-world content distribution systems have not been examined fully with respect to a number of significant performance metrics such as average download completion times and peer dynamics. Hence, a question is raised: which nodes in the network have to carry out coding operations or how does one use most of network capacity at a nominal cost with respect to network coding resources. In response this question,
a minimal set of nodes need to be selected to carry out coding operation, and this has proved to be an NP-hard problem. In this chapter, the above stated problem of minimizing network coding resources is called a Network Coding Minimization Problem (NCMP). Some attempts have been made already by using different methods to solve this problem.

Two minimal approaches have been reported in (Langberg et al. 2006 and Bhattad et al. 2005), in which an effort has been made to determine the minimal set of nodes to attain a maximum possible target rate. The approaches in (Langberg et al. 2006 and Bhattad et al. 2005) determine the minimal set of nodes to perform the coding operation by removing links in a greedy manner. A linear programming technique was reported in (Bhattad et al. 2005) to optimize the various resources utilized for network coding. Its best possible formulations involved a number of variables and constraints that increase exponentially with the number of users increased in the network. This makes it hard to apply the formulations in case of a large number of nodes, even at the price of forfeited optimality. GA have been proved to solve the diverse NP-hard problems (Holland 1975; Eiben & Smith 2003 and Yang et al. 2010), including network optimization and resource assignment problem (Rose & Yates 1996 and Mendes et al. 2009). The first attempt to apply GA to network coding was made by Kim et al. (Kim et al. 2006). Their work was extended from acyclic networks to cyclic networks and also from centralized one to decentralized one further in (Kim 2007). The genetic representation was the focus in their later work (Kim 2007), followed by a distributed algorithm to enhance GA computational efficiency (Kim 2007). Instead of tackling this NP-hard problem, the focus is to find optimal solution quickly. Finding a good order of path traversal out of many possible sequences is vital to the quality of the solutions. The problem of finding where to perform the coding operation requires selection from many numbers of choices. The proposed method manages a set of candidate solutions which are suitably less in number and enhance solutions in an evolutionary manner. This
chapter aims to propose an efficient GA based approach to solve the NCMP problem and the contributions made in this chapter are twofold.

- A problem specific genetic algorithm based approach to reduce the number of nodes to perform coding operation is proposed for decreasing the complexity involved in network code based P2P content distribution and at the same time for realizing the benefits of the network coding technique.
- GA’s performance heavily relies on the details of its elements such as population initialization, crossover operation, mutation operation, etc. Theory to accurately calculate which combination of such elements is suitable to a specific problem is not yet available. Hence, different compositions and several options of those elements which work well in many other problems are considered so as to pick the one that works best for the proposed node selection problem.

The rest of this chapter is organized as follows. Section 4.2 of this chapter gives an overview of NCMP. In section 4.3, the overview of GA is discussed briefly. Section 4.4 presents the fitness function formation for a GA based method to solve the proposed NCMP. The different methodology implemented in the proposed GA is discussed in section 4.5. In section 4.6, the main experimental results and analysis are provided and finally section 4.7 concludes the discussion.

### 4.2 NCMP OVERVIEW

RLNC is used in the proposed system. The coding complexity of the system is described as $O(ECB)$ where $E$ is the number of nodes out of $P$ nodes in the system to perform the coding operation. $B$ is the number of original blocks to be distributed, and $C$ is the size of the original content. Since $C$ is fixed for a given content and $B$ cannot be a low value, it is important to minimize the
number of coding nodes to reduce encoding complexity. Nodes in the P2P network form a directed overlay topology and it is defined as directed graph \( NW = \{P, L\} \) where \( P \) represents the set of peers or nodes and \( L \) is the set of directed overlay links which connects the peers.

The following P2P content distribution based on network coding is used

- Network topology \( NW = \{P, L\} \),
- Source in \( P \) with a content of size \( FS \) to be distributed to all peers in the network, and
- Number of coding nodes in the network \( E \) \( (1 \leq E \leq |P|) \),

The problem is to find the optimal node in the network topology where it is possible to perform the coding operation in order to shorten download time and avoid minimizing flooding of useless coded packets. Since the problem is proved as NP-hard (Xing et al. 2010), a heuristic algorithm is aimed to exploit the characteristics of the network topology. GA is a kind of stochastic search algorithm. It operates on a set of possible solutions that are improved sequentially by using procedure inspired by biological evolution (Mitchell 1996). In this chapter, the attempt is made to use GA to solve the above stated problem.

### 4.3 GENETIC ALGORITHM OVERVIEW

GA is a kind of stochastic search algorithm. It operates on a set of possible solutions that are improved sequentially by using procedure inspired by biological evolution (Mitchell 1996). GA has been used in various scientific and engineering problems, including multi objective optimization problems in networks (Yen et al. 2011). The following are the reasons for selecting GA-based approach for the proposed optimization problems.
1. GA is a promising method if the search space is large and search space of the problem is $P_L$. It grows exponentially with an increase in value P.

2. When P is sufficiently larger, it is not critical that the solution to the problem may not be a global optimum

GA starts by selecting a set of feasible solution randomly from the population. Each solution in the population is called a chromosome. Each chromosome in the population is evaluated for its fitness. This fitness defines the quality of solution. The new set of off springs is produced from the chromosomes by using operators like crossover and mutation. Each generated off springs are evaluated and most fit off springs are moved to the next generation. The less fit candidates get lesser chance because GA works on the principle of Darwin theory of evolution, which talks about the “survival of fittest”. This process is repeated until the candidates have better feasible solution to the given problem or predefined number of iterations met. Better results are discovered by repeating this process for so many iterations. GA has widely been adopted to solve many engineering problems which are difficult to solve by using traditional methods. It’s performance relies on the details of its elements such as population initialization, crossover operation, mutation operation, etc. Theory to accurately calculate which combination of such elements is well suitable to a specific problem is not yet available. The basic steps and general procedure of GA has been described next in this section. Figure 4.1 shows the general procedure of GA and Table 4.1 describes the basic notations of GA.
Figure 4.1 Basic steps in genetic algorithm

GA Notations:

Table 4.1 GA Terms and its description

<table>
<thead>
<tr>
<th>GA Terms</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chromosome (String, individual)</td>
<td>Candidate Solution (encoding the solution)</td>
</tr>
<tr>
<td>Genes (Bits)</td>
<td>Part of candidate solution</td>
</tr>
<tr>
<td>Locus</td>
<td>Position of gene in the solution</td>
</tr>
<tr>
<td>Alleles</td>
<td>Value of gene</td>
</tr>
<tr>
<td>Phenotype</td>
<td>Decoded candidate solution</td>
</tr>
<tr>
<td>Genotype</td>
<td>Encoded candidate solution</td>
</tr>
<tr>
<td>Crossover probability</td>
<td>Probability of a pair of chromosomes will be crossed</td>
</tr>
<tr>
<td>Mutation probability</td>
<td>Probability of a gene on a chromosome will be altered randomly</td>
</tr>
</tbody>
</table>
General Procedure

Procedure - Genetic Algorithm

Input : problem data, GA parameters

Output : Optimal solution

Begin

{ 
  t = 0;
  Initialize Parent (t);
  Evaluate Parent (t);
  While (Criterion not yet met)
  {
    Parents(t) = Select_Parents(P(t));
    Offspring (t) = Procreate(Parents(t));
    Evaluate(Offspring(t));
    P(t+1)= Select_Survivors(P(t),Offspring(t));
    t = t + 1;
  }

End

4.4 GA BASED NCMP – DESIGN

4.4.1 Fitness Function Formation

An algebraic formulation proposed in (Koetter & Medard 2003) for general network coding problem can be used in case where network coding is done only in a small portion of the network. Since the choice in selection of coding at a node based on the examination of its entire outgoing and incoming links, and the number of coding links is the exact measure of the amount of computation incurred by coding, the objective is to minimize the number of coding links and coding nodes. Peer with multiple incoming and outgoing links are called Likely to be a Coding Peer (LCP). It is necessary to find whether it is
possible to reduce the given peer’s output to rely on a single input without reducing the achievable download rate. This can be done by verifying the polynomials of a binary matrix B_x which is created based on the coefficients related to the incoming links of each LCP (Kim et al. 2006). B_x is feasible if the coding scheme described by it, can attain a target download rate A_T at every peer. It provides enough information to calculate the number of coding nodes and links in static network. But the initially calculated possible download rate may not be assured to achieve due to the dynamic nature of P2P network. It is necessary to calculate the new achievable download rate when there is a change in network topology and the coding plan is completely re optimized. Hence B_x is not sufficient to use in the proposed dynamic minimization problem. The problem can be devised mathematically as follows

\[
\max_{a(p,q,r)} F, \quad p=1,\ldots,m, \quad q=1,\ldots, m_{\text{out}}, \quad r = 1,\ldots,m_{\text{in}}(j) \quad (4.1)
\]

Subject to the current topology of the network at time t, G(P(t), L(t), t),

\[
D_{\text{out}} = \sum u(p,q,r)D_{\text{in}}(p,r), p = 1,\ldots,m(p), q = 1,\ldots,m_{\text{out}}(p) \quad (4.2)
\]

\[
F = \begin{cases}
    a_1 \min(A_r(p) + a_3 \text{avg}(A_r(p)) + a_2 \frac{1}{P_r + 1} + a_4 \frac{1}{P_r + 1}), & \min(A_r(p)) < A_r \quad (4.3) \\
    a_1 \min(A_r(p) + a_3 \text{avg}(A_r(p)) + a_2 \frac{1}{P_r + 1} + a_4 \frac{1}{P_r + 1}), & \min(A_r(p)) \geq A_r
\end{cases}
\]

\[
\min(a_1, a_2) > \max(a_3, a_4), \quad (4.4) \\
\min(a_5, a_6) > \max(a_1, a_2) \quad (4.5)
\]

In the above equation, F is the objective function with a_i, i = 1,\ldots,6, as specific coefficients to choose the contribution terms in F (objective function). m_{\text{in}}(p)/m_{\text{out}}(p) represent the number of incoming and outgoing paths of peer p. The data packet D_{\text{in}}(p,q) and D_{\text{out}}(p,q) are those on the q^{th} incoming and outgoing links of peer p. The weights u(p,q,r), r = 1,\ldots,m_{\text{in}}(p), determine how to combine the m_{\text{in}}(p) incoming data of peer p to generate a coded packet for the q^{th} outgoing link of peer p. A sufficient number of finite discrete values for u(p,q,r) can
assure that the maximum possible throughput is obtained in terms of download time and system throughput. A set of \( u(p,q,r) \) defines how each peer in the network forwards, replicates and/or generates the coded data. Hence, the focus in this chapter is to find the optimal value of \( u(p,q,r) \) from a finite set. A linear coding scheme determined by \( u(p,q,r) \), \( A_T(p) \) is the actual download rate achieved at peer \( p \). The optimization of coding scheme is subject to network topology defined by \( G(P(c), L(c), c) \). The weights \( u(p,q,r) \) need to be optimized in the NCMP. The objective function defined, firstly attempts to maximize the overall actually achieved download rate and after the target download rate is attained, then focus of the optimization changes to minimize the number coding nodes and coding links in the network. The terms \( \min(A_T(p)) \) and \( \text{avg}(A_T(p)) \) in the equation (4.3) are used to verify the download rate that is actually attained. A larger value for \( \text{avg}(A_T(p)) \) is expected. The term \( \min(A_T(p)) \) is used to calculate how evenly \( A_T(p) \) is optimized. Whilst as reflected by the terms \( 1/(P_n+1) \) and \( 1/(P_l+1) \) in which the network coding links and nodes are minimized. The coefficients \( a_1 \) to \( a_6 \) satisfy the conditions (4.4) and (4.5) play an important role in changing the focus of optimization from the actually achieved download rate to minimization of network coding resources. The objective function makes it possible to estimate the \( A_T(p) \), \( P_n \) and \( P_l \) in the network.

4.5 GA BASED NCMP - METHODOLOGY

4.5.1 Representation of Candidate Solutions

In GA mechanism, a candidate solution \( s \in S \) is called chromosomes. Each chromosome is made up of distinct units called genes. Each gene controls one or more features of the chromosome. In the original implementation of GA, genes are assumed to be binary numbers. This process is called mapping mechanism between the solution space and the chromosomes. This mapping is called a representation of candidate solution or encoding (Huanlai Xing et al. 2014). The representation of candidate solutions is important for designing
efficient and effective evolutionary algorithms in various difficult optimization problems, including the recently emerged coding resource minimization problem concerned with the proposed work. The problem is similar to ordering problems, such as traveling salesman problem or task ordering problem. There are several methods to represent the candidate solutions for ordering such as one-dimensional binary code, tree code and sequence and topology encoding such as ST encoding (Yen et al. 2011). These representations have low efficiency since they require search space and complexity rise with increase of nodes in the network (Tseng et al. 2006). Hence, the permutation encoding method is used to represent the candidate solutions. With this method of representing the candidate solutions, every chromosome is a string of numbers that represent a position in a sequence.

The proposed chromosome structure is based on relative information flow on path record packet flow on this path, path’s positions in the network, relative information flow on this path and incoming path’s state. This new coding scheme to represent solution should craft it possible to estimate the actually achieved download rate $A_T$ at each peer for objective function. A straight forward approach is necessary to ensure absolute information flow on path to code each solution. It is very difficult to use absolute information flow in the evolutionary approach since it causes serious problems during evolutionary operations. This means any change in the information flow on a path caused by evolutionary operations could ensure unchanged information flow on some other paths infeasible. Hence it is proposed to record relative information flow in candidate solution instead of absolute information flow. Let the integer $d$ be considered where it is interpreted as a certain predefined combination of information on incoming paths of a node. This value is also incorporated into the candidate solution representation.
It is noted that the new chromosome structure relies on topological knowledge of the entire network, which may change often in NCMP. Hence, the proposed GA cannot achieve robust performance defined by RLNC. Since the new chromosome structure records whether an incoming path will contribute to a coding occurrence or not, it may deliver robust performance. Therefore, when non contributing incoming links are broken, it is not necessary to run the optimization again. Because the proposed new structure records exact information flow, it is robust against any change in non contributing incoming paths and also if a contributing path is detached, it is easier to verify whether or not there is any non contributing path which can substitute for the broken path. The information flow to an LCP should be spread as much as possible to allow many different options in coding operation to help spread a generation. For a LCP with multiple outgoing paths, there is a high probability that the outgoing paths have different states. The following steps are used to improve up to one gene in a chromosome. By performing the following steps repeatedly, more genes can be enhanced in the chromosome. Hence, the optimal solution can be obtained quickly and it requires lesser number GA runs. Figure 4.2 represents the proposed GA run to NCMP.

**Procedure**

**Step 1** To obtain the approximate rate of information flow on each path which is associated with the chromosome representation.

**Step 2** To verify if there is an unmarked LCP, this has at least two incoming links sending the same packet. If there is no unmarked LCP, end.

**Step 3** For the present LCP, among its incoming paths which have the same packet, ensure if there is at least one path which may receive a different packet from those that the present LCP has already obtained. If there is no such path, mark the present LCP, and then go to the previous step.

**Step 4** For the present LCP, among its incoming paths which have the same packet, randomly choose a path which can convey new flow to the present LCP.
Randomly assign such a new flow to that path, alter the related gene in the chromosome and then terminate.

**Figure 4.2 Proposed GA to NCMP**

### 4.5.2 Population Initialization

Population initialization is a vital task in GA since it influences the convergence speed and also the quality of the optimum final solution. If there is no information available about the solution, then random initialization is the commonly used technique to produce initial population. The success of GA depends on the individuals chosen in the initial population and the size of the population. If poor individuals are picked in the initial population, it will finish in longer execution time and weaker optimal solutions. Since the priori information about the solution to the problem is not clearly available, the concept of opposition based optimization techniques is used for selection of the initial population. It helps in obtaining near optimal starting candidate solutions.
The following steps are carried out for opposition-based population initialization algorithm of the given problem which can be used instead of a random initialization:

1. In the first step, uniformly distributed random population is generated, IP(ps); ps is the size of population;
2. In second step, opposite population OP(ps) is calculated. The \( i^{th} \) corresponding opposite individual for OI(ps) is calculated by

\[
OI_{i,j} = c_j + d_j - P_{i,j}, \quad i = 1, 2, \ldots, ps; \quad j = 1, 2, \ldots, V_d, \quad (4.6)
\]

where \( V_d \) is the number of variables or dimension of the problem; \( c_j \) and \( d_j \) represents the interval boundaries of \( j^{th} \) variable \( (y^j \in [c_j, d_j]) \);
3. Finally, ps fittest individuals from set the \{IP(ps) U OI(ps)\} are selected as initial population.

### 4.5.3 Parent Selection and Crossover scheme

Selection is the important step in a GA in which individual genomes are selected from a population for later reproduction. An elitist model is adopted in the proposed GA for selection of parents. The best individuals are selected and they directly copy it to the next generation. The roulette wheel selection mechanism to select the rest of individuals for each generation is used. The probability of parent selection \( PR_i \), represented by \( \text{prob}(PR_i) \), is given as

\[
\text{prob}(PR_i) = \frac{F(PR_i)}{\sum_{k=1}^{ps} F(PR_k)} \quad (4.7)
\]

The better individual has the higher probability of being selected as a parent according to the definition of fitness function. Hence, the common links between two parents represent the good individuality. Let \( PR_1 \) and \( PR_2 \) be the chosen parents. The crossover operator produces child \( P_c \) by identifying the common link between \( PR_1 \) and \( PR_2 \), and retaining these links as same links in child \( PR_c \). The parameterized uniform crossover operator is used to work better
for the problem than conventional crossover operators. The correlations between coding points need not necessarily be related to the proximity between their corresponding components in a chromosome. Therefore, parameterized uniform crossover will not cause any feasibility problem.

**Procedure Roulette Wheel Selection**

**Do**

Set ps as 50

Find cumulative fitness value, total fitness value (ps) and sum of proportionate fitness value (sum)

Rotate the Wheel ps times

If sum<d then

Select the first chromosome, otherwise, select n\(^{th}\) chromosome

End if

**Until** size of the population <ps

Return chromosome with fitness value proportionate to the size of selected wheel section

**End**

**4.5.4 Evaluation of Chromosomes**

Each chromosome is created either by initialization or generated by the evolutionary operators. It is assigned a fitness value to determine how well its corresponding solution matches to the problem, based on the optimization objective function of Eq. (4.1). Here, the chromosome’s fitness value is defined as the corresponding collective actual throughput achieved. If this value is larger, then the chromosome is fit to the optimization problem.

The simple way to compute the fitness is to estimate fitness from limited trial. Suppose the fitness gain is the mean of fitness for each of the trails as calculated, the approximate of mean becomes correct. The fitness estimation
is going to be utilized for picking the members of the population. It is good that this fitness can be calculated to be a useful level of accuracy long before the intact entourage of fitness cases has been evaluated. The estimate of fitness is the average of the individual fitness cases examined so far. It is assumed that there are $I_c$ of them, the trail mean is given by

$$F' = \frac{1}{I_c} \sum_{j=1}^{I_c} F_j$$  \hspace{1cm} (4.8)

And the trail difference by

$$t^2 = \frac{1}{I_c - 1} \sum_{j=1}^{I_c} (F_j - F')^2$$  \hspace{1cm} (4.9)

given that several independent estimates are going to be added. Assume that the distribution of estimates of fitness is normal such as $D(\beta, \alpha)$. Then the trivial subsequent distribution of the mean is a scholar distribution with mean $F'$, variance $t/C$, $C-1$ degree of freedom. A scholar distribution is the distribution of the random variable $Rv = \frac{F' - \alpha}{\sqrt{C - 1}}$, which is a method of estimating the mean of the normal distribution that does not depend on the variance. The collective mass function can be utilized to test the acceptance of the estimate. Once the trial mean is adequately closer to the mean, examining fitness cases can be discontinued, and the estimate can be utilized in the reproduction phase.

### 4.5.5 Mutation Operation and Termination Criterion

When a new child is generated by applying the crossover operator, the mutation operation is done according to the mutation probability. Mutation adjusts one or more gene values in a chromosome from its early state. The solution may change completely from the previous solution in mutation. Therefore, GA can move towards better solution by using mutation operator. Mutation takes place during evolution process according to the user defined mutation probability which should be set low. The straightforward binary
mutation is used, where each bit in each chromosome is flipped autonomously with probability 0.01. First, the mutation procedure randomly selects a subset of coding nodes and makes the nodes as non-coding nodes in the network by removing all the links that connect these selected nodes and their outermost node on P. The new generation successively replaces the current one in each iteration of evolutionary process. When a pre-specified generation number is reached or when successive generations bring normalized improvements in actual throughput i.e. is all smaller than a fixed threshold value, the proposed GA is terminated.

4.6 PERFORMANCE EVALUATION

4.6.1 System Configuration

There are two parts in the implementation: GA, and network simulation. Java Genetics Algorithms Package (JPAG) is used for implementing the GA portion of the deployment, including the fitness function, recombination, and mutation. It is a Genetic Algorithms and Genetic Programming component offered as a Java framework. It offers basic genetic mechanisms that can be easily applied to evolutionary principles to problem solutions. It produces a trace of the node placement and collects the coverage data in the simulation. This data is transformed into traffic-pattern and node-placement files readable by NS2. NS2 can visualize the node placement in an animator and locate the node to perform the coding operation in P2P network. The proposed GA is converged to near-optimal solutions and convergence time is set to be 2 seconds.

Since the code minimization in P2P network is a relatively new application of genetic algorithms, little work has been reported so far. The performance of the proposed GA based Node Selection (GANS) is assessed by comparing the actually obtained network throughput in Random Selection of Nodes (RSN) to perform coding operation and Coding at All the Nodes (CAN)
in the network. In the experiments on the NCMP, there are two sets of test cases taken from (Kim et al. 2006) and modified for comparative analysis. The networks in NSet-I are generated by the algorithm in (Melancon & Philippe 2004), which connected acyclic directed graphs are randomly created. Two networks such as NSetC1 and NSetC2 are considered and utilized for simulation in NSet-I. There are 30 nodes, 90 links, 20 receivers and rate 5 in NSetC1 and 50 nodes, 130 links, 30 receivers and rate 4 in NSetC2. Table 4.2 shows the value of the various parameters used in the proposed GA.

4.6.2 Simulation Setup

GA parameters

Table 4.2 Parameters used in the proposed GA

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population size</td>
<td>50</td>
</tr>
<tr>
<td>Crossover Probability</td>
<td>0.8</td>
</tr>
<tr>
<td>Mutation rate</td>
<td>0.01</td>
</tr>
<tr>
<td>Maximum Number of Generation</td>
<td>100</td>
</tr>
</tbody>
</table>

The following table represents the parameter taken for experimental setup.

Table 4.3 Experimental parameters

<table>
<thead>
<tr>
<th>Configuration Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>File size</td>
<td>1 GB</td>
</tr>
<tr>
<td>No. of blocks in each file</td>
<td>32</td>
</tr>
<tr>
<td>Block size</td>
<td>512 KB</td>
</tr>
<tr>
<td>Maximum no of peers</td>
<td>150</td>
</tr>
<tr>
<td>Peer lifetime</td>
<td>50 or 100 s</td>
</tr>
<tr>
<td>Maximum number of uplinks</td>
<td>5</td>
</tr>
<tr>
<td>Maximum number of downlinks</td>
<td>5</td>
</tr>
<tr>
<td>Bandwidth of each up/downlink</td>
<td>256 kbps</td>
</tr>
</tbody>
</table>
4.6.3 Performance Evaluation Metrics

The following metrics are used for evaluating the performance of the proposed system:

1. **Packet redundancy**: This metric is the ratio between the total amount of network traffic and minimum amount of network traffic in the ideal case. Ideally, only one copy of the original file flood in a network and the downloading peers can collect and then reconstruct the original file through cooperation.

2. **Average packet distribution time**: It is measured by the average time all the participant peers take to complete downloading of enough coded packets to construct the original file.

3. **Maximum download time**: It is measured by the maximum amount of time for a peer to complete its download.

4. **System throughput**: It is defined as the number of bytes reconstructed successfully by all the peers per second.

5. **Failure rate**: It is referred as the number of peers unable to finish their download due to missing of some blocks.

4.6.4 Result and Discussion

The proposed mechanism performance in NSet-I and NSet-II is reported in Figures 4.4 and 4.5. Figures 4.4(a) and 4.5(a) show the download time of the previously mentioned three different techniques in NSet-I and NSet-II for different file size. It is proved that GA based node selection reduces the download time when compared to other two different techniques in both network topology. Since optimal node selection is carried out, download time is reduced in GA based approach. Figures 4.4(b) and 4.5(b) show how the proposed GA based approach resists against peer dynamics. In order to simulate this, some links are made as dynamic ones. The randomization introduced by network coding makes each packet equally important. The coding operations are carried
out and performed in the optimal nodes which make the system robust against link failures. Figures 4.4(c) and 4.5(c) prove that the proposed system reduces packet redundancy when compared to other systems even when file size increases. System throughput is increased and it remains the same in the first 400 minutes in all those three techniques. There is a slight improvement in GA based technique after 400 minutes in system throughput. It is reported in Figures 4.4(d) and 4.5(d). The Download time in Figure 4.3 (a) and Figure 4.4 (a) is represented in seconds.

Figure 4.3 (a) Download time vs file size

Figure 4.3 (b) Failure rate vs number of dynamic links
Figure 4.3 (c) Packet redundancy vs file size

Figure 4.3 (d) System throughput
Figure 4.3 NSet - I – performance analysis in NsetC1

Figure 4.4 (a) Download time in different runs
Figure 4.4 (b) Failure rate vs number of dynamic links

Figure 4.4 (c) Packet redundancy vs file size
Network coding is a proven technique to solve rarest block problem in P2P network and it can also be used in network protocol, wireless networks, and network security. It is necessary to minimize the coding operation carried out in network code based P2P content distribution due to the complexity involved. The optimization of network coding aims to reduce network coding resources such as coding nodes and links. It has recently attracted research attention. This chapter considered how to address the NCMP by describing its general formulation, and then by developing an efficient GA to realize it. GA is used to find the optimal node to perform coding operation in the network code based P2P network in order to achieve optimization of merits and demerits of network coding. Its performance relies on its various elements such as population initialization, crossover operation, and mutation operation. In this proposed GA, problem specific GA operators such as opposition based technique for population initialization and parameterized uniform crossover operator are used. Both operators work better for the proposed code minimization problem. The proposed method is compared with two different
techniques by focusing on mainly download time and redundant coded packet transmission. The simulation outcome proves that the proposed method works better in terms of download time and packet redundancy rate as well as failure rate.