Enhancing Data Transmission Reliability with Multipath Multicast Rate Allocation
Matin Bagherpour, Mehrdad Alipour and Øivind Kure

Abstract—In this paper, a multipath routing scheme is proposed for data transmission in a packet-switched network to improve the reliability of data delivery to multicast destinations, and to reduce network congestion. A multi-objective optimization model is presented that utilizes FEC (Forward Error Correction) across multiple multicast trees for transmitting packets toward the destinations. This model assigns the transmission rates over multicast trees so that the probability of irrecoverable loss for each destination and also the link congestion is minimized. We propose a genetic algorithm based on SPEA (Strength Pareto Evolutionary Algorithm) in order to approximate Pareto optimal solutions of this rate allocation problem with a non-dominated solution set. Numerical results show that splitting data packets between multiple trees increases reliability and decreases network congestion when compared with the results obtained for transmitting data packets over a single tree.

Index Terms—Forward Error Correction (FEC), load balancing, multicast communication, multipath routing, Quality of Service

I. INTRODUCTION

PATH diversity can be achieved by setting up multiple parallel paths between source and destination nodes. Multipath routing not only can reduce congestion in the network, but also can be considered as a tool for error resilience by providing higher bandwidth for each session. The idea of using multiple parallel paths for transmitting data was first proposed in [1]. In this work, a message is divided into a number of sub-messages and the sub-messages are transmitted over disjoint paths in the network.

A comprehensive review of multipath routing for load balancing and traffic engineering considering Quality of Service (QoS) is presented in [2]. The authors introduced a general multi-objective optimization model to balance traffic load among multiple trees and optimize QoS measures such as average delay, and average delay jitter.

Since transport protocols such as TCP favor reliability over timeliness, they are not appropriate for real-time streaming applications. Therefore, many approaches have been proposed to deal with these kind of applications. Layered and error-resilient video coding are two approaches of this kind. Layered video codec adapts the internet bit rate to the available bandwidth and tries to deal with time-varying nature of the internet [3]. In error-resilient codec, the bit stream is modified in a way that the decoded video degrades more smoothly in lossy environments [3]–[5]. It is shown that multipath transmission, when combined with error control schemes, can improve the quality of multimedia services in terms of packet loss and delay. There has recently been an increasing interest in using multipath routing for failure recovery of real-time multimedia applications. For example, Multiple Description Coding (MDC) and Forward Error Correction (FEC) are combined with multipath routing to improve data transmission in internet. In MDC approach, a video source is split into multiple descriptions and each of them is sent over a different channel. MDC has been studied in detail in [6]–[10]. FEC is a channel coding technique which increases reliability at the expense of bandwidth expansion [11]–[14].

There are also some approaches based on multicasting to stream multimedia sessions over the internet [15]. Multicast reduces bandwidth consumption by not sending duplicate packets on the same physical link of the network [16].

In this paper, we utilize path diversity over packet switched networks in transmitting data packets from a source node to multiple destination nodes of a multicast group. We integrate multicast routing and multipath transmission with failure recovery to improve reliability in data transmission and reduce congestion in a packet switched network. We suppose that multicast trees have the ability to send data packets from the source to destinations at different rates. In this work, each network link is modeled as a continuous Gilbert-Elliot channel as in [17]. A Gilbert-Elliot channel can have two states, namely “good” and “bad” states. During transmission of a packet, if the channel is in the good state, the packet will be delivered to the destination successfully; otherwise, it will be lost. We use FEC scheme to split data packets between multicast trees, i.e. the data is encoded into a block of equal packets so that each destination is able to recover the video session by receiving at least K packets (K ≤ N); otherwise, an irrecoverable loss happens. A multi-objective optimization model is presented for the aforementioned problem which tries to minimize the probability of irrecoverable loss (reliability maximization) and network congestion (by minimizing maximum utilization of network links).

By this model, we try to exploit both load balancing and failure recovery advantages of path diversity in transmitting data packets to receivers. We use simulation to estimate probability distribution of bad burst times of each path in the network in order to calculate the probability of irrecoverable loss for each destination. Since the multi-objective model is highly nonlinear and cannot be solved by common solvers, a genetic algorithm based on Strength Pareto Evolutionary
Algorithm (SPEA) is presented to solve the multi-objective model for a sample network topology.

The rest of this paper is organized as follows. Related works are surveyed in this section and motivation of this work is discussed. In Section II, the network model is introduced. Our mathematical programming model of the problem is given in Section III. In Section IV, the proposed genetic algorithm based on SPEA is presented. In Section V, numerical results of implementation of the algorithm for a sample network are presented and the advantages of using multiple trees over single tree are illustrated. In Section VI, conclusions and suggestions for future research are given.

A. Related Works

In previous works, multipath routing is combined with MDC in order to enhance resilience to loss in video streaming [18], or to reduce rate distortion of video streams [19]. Multipath routing of TCP packets is used to control the congestion in networks with minimum signaling overhead [20]. Path diversity is also used over IP voice streams to increase speech quality [21]. In addition, the problem of rate allocation over multiple paths is presented in [22] and [23]. In [22], the authors consider a leaky bucket model for network paths and try to minimize the end-to-end delay. The rate allocation problem with multiple senders to a single receiver is presented in [23]. The authors suppose that the connection between each pair of source and receiver is a Gilbert-Elliot channel and propose an algorithm to solve packet partitioning and rate allocation problems. A rate allocation algorithm is used to minimize probability of irrecoverable loss in FEC approach.

FEC is a system of error control for data transmission, where the sender adds redundant data to the messages to increase the chance of successful data recovering at the receiver. An irrecoverable loss occurs if the number of successfully delivered packets is smaller than the number of initial packets before adding redundant packets. Since complexity of the proposed model in [23] depends on the number of packets and increases exponentially by the number of paths, the authors proposed a brute-force search algorithm to solve the rate allocation problem for the special case of two disjoint paths. They considered disjoint paths to facilitate modeling of irrecoverable loss, but it should be noted that finding completely disjoint paths may only be possible in highly connected networks. The advantages of their approach in reducing probability of irrecoverable loss are investigated by implementing it for the actual internet in [24].

A rate allocation problem is also presented in [17] where the authors take advantage of path diversity to send data packets from the source to destinations over general packet switched networks, like internet, in order to minimize the probability of irrecoverable loss. They assume that the paths are disjoint, and each path is modeled as a continuous Gilbert-Elliot channel. The authors calculate probability of irrecoverable loss by using a continuous approximation for probability distribution of the time each path spends in the bad state during a block time. However, in order to simplify calculation of probability distribution of bad state time, they assume that each path cannot have more than one bad burst time during a block time.

II. NETWORK MODEL

A. Link Model

We model the network links with a two-state continuous time Markov process: Gilbert-Elliot. According to the Gilbert-Elliot model, a channel spends an exponentially distributed amount of time with mean $1/\mu_g$ in the good state. Then, it alternates to the bad state and stays in the bad state for another exponentially distributed amount of time with mean $1/\mu_b$. Although this model is used for network paths in [17], we make use of it for each link in the network. This assumption is justified by the fact that the Gilbert-Elliot model has the ability to model a single transmission channel whereas network paths usually consist of several links and cannot be considered as single transmission channels. On the other hand, independent Gilbert-Elliot channel model is only applicable for disjoint paths. In this paper, each path consists of several links that each is modeled as an independent Gilbert-Elliot channel. It is also assumed that the good time mean $1/\mu_g$ of a link is much greater than its bad time mean $1/\mu_b$, and the channel state does not change during transmission of a packet [17]. If a packet is transmitted during the bad state of a link, it will be lost before reaching the downlink node; otherwise, it will be delivered to the downlink node successfully. This model is widely used for transmission channels for the applications where delay is not a critical factor [17].

B. Error Correction Model

In this work, FEC is applied across multiple multi-rate trees to reduce probability of irrecoverable loss occurrence for each destination. In this scheme, data is encoded into a block of $N$ equal packets so that each destination can recover the whole data by receiving at least $K$ packets. $K$ represents the number of initial packets before adding redundant data packets. In other words, an irrecoverable loss occurs for a destination if at least $N-K$ packets are lost out of $N$ packets that are transmitted toward that destination. By modeling network links as Gilbert-Elliot channels, we are able to formulate the irrecoverable loss for non-disjoint multicast trees for transmitting data packets to multicast destinations. Mathematical optimization formulation of the aforementioned problem is presented in section III.

III. MATHEMATICAL FORMULATION

In this model, $MT$ represents the set of multicast trees, $M$ the set of destinations, and $L$ the set of network links. It is assumed that each multicast tree has the ability to transmit packets from the source to all the destinations in $M$ at different rates. The problem is formulated as a multi-objective mathematical programming model as follows:

$$\min P_E = P\left(\sum_{i=1}^{\vert MT\vert} B_{ij}X_{ij} \geq N-K\right) \forall j = 1, \ldots, \vert M\vert$$  (1)
TABLE I
NOTATIONS: PARAMETERS AND DECISION VARIABLES

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Total number of packets to be sent to each destination;</td>
</tr>
<tr>
<td>K</td>
<td>Minimum number of data packets needed in each destination to recover the whole data (number of information packets);</td>
</tr>
<tr>
<td>T</td>
<td>Total block time;</td>
</tr>
<tr>
<td>path$_{ij}$</td>
<td>Path associated with the jth destination in the ith multicast tree;</td>
</tr>
<tr>
<td>B$_{ij}$</td>
<td>Random variable representing the portion of time at least one of the links of path$_{ij}$ spends in the bad state during the total block;</td>
</tr>
<tr>
<td>P$_{E}^{ij}$</td>
<td>Probability of irrecoverable loss for the jth destination;</td>
</tr>
<tr>
<td>$C_{uv}^{\text{max}}$</td>
<td>Maximum allowable capacity of the link $(u, v)$ for data transmission;</td>
</tr>
<tr>
<td>K'</td>
<td>Maximum number of paths to transmit packets for each destination;</td>
</tr>
</tbody>
</table>

Decision variables

$$\min \max_{(u,v) \in L} \left( \sum_{j=1}^{[MT]} \left( \frac{X_{ij} \lambda_{ij}^{uv} T}{C_{uv}^{\text{max}}} \right) \right)$$  \hspace{1cm} (2)

$$\sum_{i=1}^{[MT]} X_{ij} = N \quad \forall j = 1, \ldots, [M]$$  \hspace{1cm} (3)

$$\sum_{i=1}^{[MT]} \max_{j=1}^{[M]} \left( \frac{X_{ij} \lambda_{ij}^{uv} T}{C_{uv}^{\text{max}}} \right) \leq C_{uv}^{\text{max}} \quad \forall (u, v) \in L$$  \hspace{1cm} (4)

$$\sum_{i=1}^{[MT]} \left( \frac{X_{ij}}{N} \right) \leq K' \quad \forall j = 1, \ldots, [M]$$  \hspace{1cm} (5)

The model parameters and decision variables are defined in Table I.

The objective is to minimize maximum utilization of network links. This objective is widely used in traffic engineering optimization to minimize network congestion. A survey of the objective functions and constraints of traffic engineering problems is presented in [2]. Link congestion is calculated based on that each multicast tree transmits only the maximum number of data packets sent over it toward different destinations, not the sum of data packets. This is an advantage of multicast communications [2]. In (2), we extend the maximum link utilization function for packet switched networks in which the data is transmitted as discrete data packets over network links to continuous data transmission. Continuous data transmission means that a constant bit rate of data is dedicated to network paths. Equations (3) indicate that summation of the allocated packets to the paths in different trees associated with the jth destination must be equal to the total number of packets sent toward this destination. Inequalities (4) stand for the capacity constraint of communication links. The constraint on the maximum number of paths that can be used for transmitting packets to each destination is represented by (5).

In order to calculate the probability of irrecoverable loss for each destination, we need to have the probability distribution of $B_{ij}$s. We use discrete event simulation to obtain distribution of $B_{ij}$s. The main reason that we use simulation is that analytical derivation of probability distribution of bad time portion for network paths is complicated. Even by considering unrealistic simplifying assumptions in [9], the authors could not derive these distributions in general. The details of this simulation model and simulation results are discussed in Section IV.

Since this optimization problem is a multi-objective model with integer decision variables and nonlinear relations, it cannot be solved by common optimization solvers. We propose a multi-objective evolutionary algorithm based on SPEA to solve the presented model.

IV. SOLUTION PROCEDURE

A. Simulation Study

Based on the Gilbert-Elliot model, each link spends an exponentially distributed amount of time in the good state and then, alternates to the bad state and spends another exponentially distributed amount of time there.

We define the bad state of the path path$_{ij}$ as the state that at least one of its links is in the bad state. We need to derive the distribution of the time that this path spends in the bad state, represented by $P(B_{ij})$, in order to calculate the probability of irrecoverable loss. Analytical computation of probability distribution of $B_{ij}$s is extremely complicated without making unrealistic simplifying assumptions. For example, the authors in [17] analytically obtained the distribution of the bad burst time for each path assuming that each path cannot have more than one bad burst during a block time. In this work, we relax this assumption and use simulation to find the distribution of the bad time for each path. We relax this assumption due to the fact that it is an unrealistic assumption and is considered in [17] just to facilitate the process of finding aforementioned probability distributions. In our simulation study, good and bad times for network links are exponentially distributed variables.

The probability of irrecoverable loss for the jth destination, is calculated by continuous approximation as in [17] i.e. the number of lost data packets in each path equals to the portion of time that the path spends in the bad state multiplied by the number of data packets that are transmitted over this path. This approximation is rational if we suppose that the packet inter-arrival time is much shorter than each typical bad burst of each link in the network, $\frac{1}{\lambda} << \frac{1}{\mu_{b}}$. $\frac{1}{\lambda}$ represents the inter-arrival packet time and $1/\mu_{b}$ represents the bad time mean for a network link. In this case this condition does not hold, any two consecutive packets are transmitted on two independent states of each channel. Therefore, there could be no gain by applying path diversity as compared to single path transmission [17]. The first objective function (1) minimizes the probability of irrecoverable loss for multicast destinations. The second objective function presented by (2) attempts to minimize maximum utilization of network
Means of these random variables are generated according to a uniform distribution. Simulation results will be presented in Section V.

B. Multi-Objective Evolutionary Algorithm Approach

We propose a multi-objective evolutionary algorithm to solve the traffic splitting problem modeled in Section III. A survey of the works successfully used multi-objective evolutionary algorithms for traffic engineering problems is available in [2].

Genetic algorithm is one of the widely used successful evolutionary algorithms. In the genetic algorithm, an initial population of feasible solutions is generated as the starting point of the search. Individuals are evaluated based on the value of a fitness function. The fittest individuals are selected as parents according to a selection method to generate the next generation by using genetic operators such as crossover and mutation. This procedure continues repeatedly by replacing the old generation with the new one and keeping the best individuals of each generation until a terminating condition is satisfied.

In this paper, a genetic algorithm based on SPEA [25] is proposed. In SPEA, in each generation an external set with the best Pareto solutions are held in addition to the evolutionary population. In the multi-objective optimization context, a solution $x$ from the solution space dominates another solution $y$ from that space if and only if $x$ is as good as $y$ regarding every objective, and is strictly better than $y$ in at least one objective function. In the case that neither $x$ nor $y$ dominates the other, it is said that they are Pareto solutions or non-dominating solutions in contrast to each other. In SPEA, solutions in the external set are pruned and updated in each generation whenever a solution from the population dominates at least one solution in the external Pareto set.

Our proposed multi-objective genetic algorithm is structured like SPEA with some modifications in generating the initial population and fitness assignment strategy. These changes are necessary because SPEA is constructed to solve non-constrained multi-objective problems, whereas our problem is constrained. Since generating a feasible initial population in a way that satisfies all the constraints is not trivial, and also the population will not necessarily remain feasible after using crossover and mutation operators over the individuals, we use a penalty based approach to penalize the individuals that are not feasible based on their degree of infeasibility according to [26]. In this approach, an adaptive penalty function and a distance function are used to penalize infeasible individuals to reduce their chance of being selected as parents. Penalty based approaches have a good reputation in solving constrained optimization problems with evolutionary algorithms. This method is easy to implement and does not need any parameter tuning when compared with the other available approaches.

The proposed multi-objective genetic algorithm has the following steps:

**Step 1:** The initial population $P$ is generated randomly in a way that each individual is feasible according to the constraints (3) and (5), but it can be infeasible regarding the capacity constraints represented by (4). Also, an empty set $P'$ of non-dominated solutions is created.

**Step 2:** The non-dominated individuals of Pare copied into $P'$ and the solutions within $P'$ which are covered by the other members of $P'$ are removed. The solution $x$ is said to cover the solution $y$, if and only if, $x$ dominates $y$ or $x$ and $y$ have the same fitness considering all the objective functions.

**Step 3:** If the number of non-dominated solutions exceeds a given maximum number $N'$, then $P'$ is pruned by using a clustering method to limit the size of Pareto solutions to the specified size $N'$. Clustering is used to reduce the size of non-dominated solution set to its predefined size by keeping representative solutions that have specifications of all the solutions. We use average linkage method because it has proven to perform well with SPEA algorithm [25].

**Step 4:** For each individual $x$, the values of $|M|$ objective functions as formulated in (1), and link utilization as in (2) are calculated and preserved in the vector $Obj_j$. ($Obj_j$ is a vector of $|M| + 1$ scalar values). Then, the values of objective functions are modified for each solution according to the penalty based approach. The degree of violation of capacity constraints is calculated for each individual $x$ as follows:

$$v(x) = \frac{1}{|L|^2} \sum_{j=1}^{L^2} \frac{C_j(x)}{C_j^{\max}}$$

where $C_j(x)$ represents the degree of violation of the $j^{th}$ capacity constraint for individual $x$, and:

$$C_j^{\max} = \max_x C_j(x)$$

$C_j(x)$ takes positive value if $j^{th}$ capacity constraint in equations (4) is violated by solution $x$; otherwise, it is 0.

The distance value of the individual $x$ in each dimension $i$ of objective function ($i^{th}$ element of vector $Obj_j$) is calculated as follows:

$$d_i(x) = \left\{ \begin{array}{ll} v(x) & \text{if } r_f = 0 \\ \sqrt{Obj_j(i)^2 + v(x)^2} & \text{Otherwise} \end{array} \right.$$  \hspace{1cm} (8)

where $Obj_j(i)$ represents the $i^{th}$ element of vector $Obj_j$, and $r_f$ indicates the portion of feasible individuals in the current population. $r_f$ takes value from [0,1]. If there is no feasible individual in the current population, the individuals with smaller capacity constraint violation values will have smaller distance function in the $i^{th}$ dimension and will dominate the other individuals in this dimension. If there is at least one feasible individual in the population, those feasible individuals with smaller objective function value will dominate the other individuals in the $i^{th}$ dimension. In this case, among the infeasible individuals, the ones that are closer to the origin in $Obj_j(i) - v(x)$ space will have smaller distance function regarding the $i^{th}$ dimension [26].

The penalty function for the $i^{th}$ objective function of individual $x$ is calculated according to (9):

$$p_i(x) = (1-r_f)X_i(x) + r_fY_i(x)$$  \hspace{1cm} (9)

where

$$X_i(x) = \left\{ \begin{array}{ll} 0 & \text{if } r_f = 0 \\ v(x) & \text{Otherwise} \end{array} \right.$$  \hspace{1cm} (10)
\[ Y_i(x) = \begin{cases} 0 & \text{if } x \text{ is a feasible individual} \\ Obj_j(i) & \text{Otherwise} \end{cases} \] (11)

This penalty function penalizes the infeasible individuals even more. The first part of this penalty function, \((1 - r_j)x_i(x)\), has larger values for the individuals with large amount of constraint violation and will have more effect when \(r_j\) tends to zero. In the second part of the penalty function, \(r_jY_i(x)\), the infeasible individuals with larger objective function value will be penalized more. This penalty function has more impact when \(r_j\) tends to one \([26]\).

Finally each dimension \(i\) of the modified objective function for the individual \(x\) is obtained by using (8) and (9) as follows:

\[ \text{Modified}_i\text{Obj}_j(i) = d_i(x) + p_i(x) \] (12)

**Step 5:** Fitness function of individuals in \(P\) and \(P'\) is calculated as follows:

Each individual \(i \in P\) is assigned a strength value \(s_i = n/(n+1) \in [0,1]\), where \(n\) represents the number of individuals in \(P\) which are dominated by \(i\), and \(N\) represents the size of \(P\). Fitness of each individual \(j\) is defined as follows:

\[ f_j = \begin{cases} s_j & j \in P' \\ 1 + \sum_{i \geq j} s_i & j \in P \end{cases} \] (13)

**Step 6:** The individuals are selected from \(P + P'\) as parents to form the mating pool according to their fitness. In this study, the Roulette wheel selection is used to choose the parents.

**Step 7:** Crossover and mutation operators are applied to the parents. In this work, we use flat and random operators which are proposed in [27] and [28] respectively. As a result, population of the next generation is generated.

**Step 8:** The procedure terminates if the domination rate of the current generation is zero, otherwise it is continued from **Step 2**. Domination rate in a generation is defined as the portion of the individuals in the non-dominated set of the previous generation which are dominated by the non-dominated individuals in the current generation.

In the next section, numerical results of implementation of the multi-objective genetic algorithm for a sample network are presented.

V. NUMERICAL RESULTS

A. Simulation Results

The 14-node NSF (National Science Foundation) network topology is chosen to study the performance of proposed algorithm. This network topology is shown in Fig. 1. We consider node \(N_0\) as the source of the multicast transmission and nodes \(N_4, N_5, N_9,\) and \(N_{12}\) as the destination nodes. We also consider that three multicast trees are available to transmit data packets from the source to destinations. Let \(MT = \{ T_1, T_2, T_3 \}\) be the set of multicast trees. The multicast trees \(T_1, T_2\) and \(T_3\) are illustrated in Fig. 2 by using red, blue, and green colors respectively.

As mentioned before, discrete event simulation is used to estimate probability distribution of the portion of time that each path of multicast trees spends in the bad state out of the total block time \(T\). For this purpose, we assume that the total block time \(T\) is 1 s. We consider network links with uniformly distributed random good time and bad time means, and simulate the system to observe and record the behavior of network paths in order to derive distribution of the bad time portions. The good time mean for each network link is generated according to a uniform distribution with parameters 0.9 s and 1 s, and the bad time mean of each network link is uniformly distributed with parameters 0.01 s and 0.02 s. The range of these parameters is chosen based on the previous works [17] and [23].

We replicate the simulation 1000 times for each path in order to have adequate number of observations to be able to find the shape of probability distribution functions, and estimate the parameters of the distribution. Three well-known statistical hypothesis tests are applied to data samples, namely Kolmogorov-Smirnov test, Anderson-Darling test, and Chi-squared test. The results show that the distribution that fits best with the data samples of the bad time portion of each path and successfully passes all statistical tests is the shifted Log-Normal. Data samples for each path pass the hypothesis of following shifted Log-Normal distribution. The parameters of the shifted Log-Normal distribution for each path are obtained by using maximum likelihood estimation. Values of parameters obtained for the Log-Normal distribution associated of each path are given in Table II. In this table, \(Path_{T_i,N_j}\) represents the path associated with destination \(N_j\) in tree \(T_i\).

When a random variable \(X\) has shifted Log-Normal distribution with parameters \((\mu, \sigma, \gamma)\), variable \((X - \gamma)\) has Log-Normal distribution with parameters \((\mu, \sigma)\), i.e. \(Y = \)...
Fig. 3. Data histogram of path $T_1, N_4$ and fitted Log-Normal $(\mu, \sigma, \gamma) = (-1.74, 0.124, -0.07)$

Fig. 4. Data histogram of path $T_1, N_5$ and fitted Log-Normal $(\mu, \sigma, \gamma) = (-1.701, 0.114, -0.095)$

Fig. 5. Data histogram of path $T_1, N_9$ and fitted Log-Normal $(\mu, \sigma, \gamma) = (-2.037, 0.141, -0.054)$

Fig. 6. Data histogram of path $T_1, N_{12}$ and fitted Log-Normal $(\mu, \sigma, \gamma) = (-1.098, 0.071, -0.206)$

$\log(X - \gamma)$ has normal distribution with parameters $(\mu, \sigma^2)$. Histograms of the observed bad time portions of the paths associated with destinations $N_4, N_5, N_9$, and $N_{12}$ in $T_1$ are illustrated in Fig. 3 to Fig. 6 respectively. These figures also demonstrate Log-Normal distribution curve that is fitted on data samples.

B. Genetic Algorithm Implementation Results

After obtaining probability distribution of the bad time portion of paths, we use our proposed algorithm to find solutions for the optimization problem presented in Section III. In each generation of the proposed genetic algorithm, we need to calculate the irrecoverable loss probabilities according to (1) for each individual. To be able to calculate these probabilities, we need to have the cumulative probability distribution of the weighted sum of Log-Normal variables. $P_\Sigma$ is calculated supposing that $B_{ij}$s in (1) have shifted Log-Normal distribution. Since distribution of sum of Log-Normal variables cannot be found in the closed form, we use the well-known approximation of Fenton and Wilkinson [29]. They approximated summation of Log-Normal variables with another Log-Normal variable. If we assume that $X_j$ $j = 1, ..., n$ has Log-Normal distribution with parameters $(\mu_j, \sigma_j)$, then sum of $X_j$s is approximated by a Log-Normal variable with parameters $(\mu, \sigma)$ as follows:

$$\sigma^2 = \log \left( \frac{\sum_{j=1}^n e^{2\mu_j+\sigma_j^2} (e^{\sigma_j^2} - 1)}{\left(\sum_{j=1}^n e^{2\mu_j+\sigma_j^2/2}\right)^2} \right)$$  \hfill (14)

$$\mu = \log \left( \frac{\sum_{j=1}^n e^{2\mu_j+\sigma_j^2/2}}{\left(\sum_{j=1}^n e^{2\mu_j+\sigma_j^2/2}\right)^2} - \frac{\sigma_j^2}{2} \right) \hfill (15)$$

By this approximation, fitness value of each individual in the genetic algorithm can be easily calculated from (1). We implemented the proposed genetic algorithm for the network and paths illustrated in Fig. 2 with random good and bad time for network links.

The values of genetic algorithm parameters are presented in Table. III. We run the algorithm with two values for the maximum number of paths for each destination. Also, the proposed genetic algorithm is run for 12 different capacity sets for network links.

Figs. 7-10 illustrate the average of irrecoverable loss probability versus $K$ obtained by single-tree transmission and multi-tree transmission. In multi tree case, we are allowed to use all the aforementioned trees to send the data packets toward the destinations, whereas in single tree case, we can use only one specific tree to deliver the data packets to the
destinations. Since we have multiple non-dominated solutions in each implementation of genetic algorithm, we calculate the average value of each objective function over different solutions in order to obtain one representative in each \( \mu \sigma \gamma \) implementation of the algorithm.

We can see from Fig. 7 that for NSF network topology, the average probability of irrecoverable loss for destination \( N_4 \) by using multiple trees is smaller than that by using only trees \( T_1 \) and \( T_3 \). However, it is greater than the value obtained by using only \( T_2 \). The similar observation can be seen for the other destinations in other figures.

To compare multi-tree transmission and single-tree transmission, we can see from Fig. 7 to Fig. 10 that if we use only the tree \( T_1 \) to send data packets toward the destinations, probability of irrecoverable loss for the destinations \( N_9 \) are smaller as compared to the multi-tree case. However, probabilities of occurrence of irrecoverable loss for \( N_4, N_5, \) and \( N_12 \) in single-tree case are respectively more than 0.5, approximately 0.3 and more than 0.8 that are much greater than the probability under multi-tree transmission. These probabilities of irrecoverable loss for these destinations can be considered very unacceptable. We can observe similar behavior for other destinations. In the presented results, we observe that sending all the data packets over only one tree results to enhance the probability of irrecoverable loss for some destinations, but can not necessarily keep the probability low enough for all destinations and make them much worse as compared to multi tree case.
TABLE IV  
NUMERICAL RESULTS

<table>
<thead>
<tr>
<th>Multicast trees</th>
<th>Decision variables</th>
<th>Objective functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>T₁</td>
<td>[ X_{11} = 1000, ] [ P^T_{11} = 0.4925, ] [ X_{12} = 1000, ] [ P^T_{12} = 0.1172, ] [ X_{13} = 1000, ] [ P^T_{13} = 0.1072, ] [ X_{ij} = 0 \text{ for } i=2,3, j=1,2,3,4. ]</td>
<td>P^T_{1} = 0.8641.</td>
</tr>
<tr>
<td>T₂</td>
<td>[ X_{21} = 1000, ] [ P^T_{21} = 0.0041, ] [ X_{22} = 1000, ] [ P^T_{22} = 0.1255, ] [ X_{23} = 1000, ] [ P^T_{23} = 0.4711, ] [ X_{ij} = 0 \text{ for } i=2,3, j=1,2,3,4. ]</td>
<td>P^T_{2} = 0.4980.</td>
</tr>
<tr>
<td>T₃</td>
<td>[ X_{31} = 1000, ] [ P^T_{31} = 0.1183, ] [ X_{32} = 1000, ] [ P^T_{32} = 0.5024, ] [ X_{33} = 1000, ] [ P^T_{33} = 0.4905, ] [ X_{ij} = 0 \text{ for } i=2,3, j=1,2,3,4. ]</td>
<td>P^T_{3} = 0.1180.</td>
</tr>
<tr>
<td>T₁, T₂, T₃</td>
<td>[ X_{11} = 61, X_{12} = 508, X_{13} = 977, ] [ P^T_{1} = 0.0019, ] [ X_{14} = 2, X_{21} = 692, X_{22} = 488, ] [ P^T_{2} = 0.0547, ] [ X_{23} = 19, X_{24} = 4, X_{31} = 247, ] [ P^T_{3} = 0.1071, ] [ X_{32} = 4, X_{33} = 4, X_{34} = 994, ] [ P^T_{3} = 0.1190. ]</td>
<td></td>
</tr>
</tbody>
</table>

To show that sending data packets over multiple trees can improve reliability of transmission to multicast destinations in comparison with single-tree, we implemented the model with a different parameter setting. In this test, the means of good and bad times for all network links are equal to 0.1 s and 0.01 s respectively. We relaxed the capacity constraint of links. This allows the decision variables associated with a destination take values independent from values of decision variables associated with other destinations. We used the first objective functions in (1) as the fitness function in the proposed genetic algorithm. By this setting, values of decision variables associated with each destination are independent from those of other destinations according to the mathematical model in Section III. The results of implementing the genetic algorithm for single-tree and multi-tree transmission and for \( K' = 950 \) are illustrated in Table IV. It can be observed that when we are allowed to use all the multicast trees to send data packets, probabilities of irrecoverable loss are significantly less than single tree transmission. Number of data packets transmitted over trees T₁, T₂, and T₃ are equal to the maximum number of packets that are being transmitted to different destinations over these trees and can be calculated having values of decision variables.

As mentioned before, a domination rate is calculated at each iteration of the genetic algorithm that represents the portion of Pareto solutions in that iteration that dominate the Pareto solutions of the previous iteration. Fig. 11 to Fig. 13 show the domination rate at iterations of the proposed genetic algorithm for \( K = 950, 955, \) and 960. The genetic algorithm stops when there is only one non-dominated solution remained in the non-dominated set in several subsequent iterations of the algorithm. We can see from these figures that the domination rate in the Pareto solutions at initial iterations is higher and it gradually converges to zero after around 100 generations. This rate can be pegged as a convergence sign of the genetic algorithm. Multi-tree transmission can also reduce network congestion in comparison to single-tree transmission. Fig. 14 represents the average value of maximum link utilization function versus link capacity for multi-tree and single-tree. The single-tree curve represents the average of results obtained from sending data packets over the first, the second and the third tree. We can see from Fig. 14 that the average of network congestion in multi-tree transmission is much smaller than this value for single-tree cases. It can also be inferred from this figure that when the links have lower capacity, the single tree solutions.
are not feasible considering capacity constraints, because the value of maximum link utilization is greater than 1. However, the average congestion for multi-tree solution is always below 1. These results indicate that multi-tree transmission can be the better choice when we have strict capacity constraints.

VI. CONCLUSION

In this work, we proposed a multi-objective mathematical formulation for multipath multicast rate allocation problem in order to minimize the probability of irrecoverable loss and also minimize network congestion. For calculating probabilities of irrecoverable loss for each destination, we estimated distribution of the time that each path spends in the bad state out of the total block time. Discrete event simulation was used to derive this probability distribution for NSF network topology. We observed that the bad time portion of network paths follows shifted Log-Normal distribution. We proposed a multi-objective genetic algorithm based on SPEA to solve the model and to find the Pareto solutions. Numerical results show that multipath routing significantly decreases probability of irrecoverable loss in comparison to single-tree transmission.

Deriving the relationship between parameters of Log-Normal distribution found for bad state time and number of links in each path and the parameters of Gilbert-Elliot model can be considered as a future research to the work presented in this paper. This work can further be expanded by considering other QoS measures such as end-to-end delay and delay jitter.

ACKNOWLEDGEMENTS

This work was supported by Centre for Quantifiable Quality of Service in Communication Systems, Centre of Excellence, appointed by the Research Council of Norway, and funded by the Research Council, NTNU and UNINETT (http://www.q2s.ntnu.no).

REFERENCES


Matin Bagherpour is a postdoctoral researcher in Centre for Quantifiable Quality of Service in Communication Systems (Q2S) in the Norwegian University of Science and Technology (NTNU). She received her PhD in Industrial Engineering from Tarbiat Modares University (Iran) in May 2008. She has worked as an assistant professor in University of Science and Culture (Iran) since then. Her research interests include optimization (mathematical programming, integer programming, meta-heuristic algorithms, combinatorial optimization), especially optimization of telecommunication networks (routing, scheduling, resource allocation, pricing, QoS, etc.), and ICT economics.

Mehrdad Alipour is graduated from the Graduate School in Industrial Engineering, University of Science and Culture (USC), Tehran, Iran. He received his B.Sc. degree in Industrial Engineering from the Isfahan University of Technology (IUT), Iran in 2008. His research interests include optimization in telecommunication networks, meta-heuristic optimization approaches, and stochastic programming.

Øivind Kure is a Professor at Centre for Quantifiable Quality of Service in Communication Systems (Q2S) in the Norwegian University of Science and Technology (NTNU). He received his PhD from the University of California, Berkeley, USA. His current research interests include quality of service in wired and wireless networks, sensor networks, and routing.