2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Smart Bandwidth Management using a Recurrent Neuro-Evolutionary Technique

Rabia Arshad
Department of Electrical Engineering,
NWFP University of Engineering and Technology,
Peshawar,Pakistan
Email: rabia.arshad@nwfpuet.edu.pk

Gul Muhammad Khan and Sahibzada Ali Mahmud
Department of Electrical Engineering,
NWFP University of Engineering and Technology,
Peshawar,Pakistan
Email: gk502@nwfpuet.edu.pk
and sahibzada.mahmud@nwfpuet.edu.pk

***Abstract*—The requirement for correct bandwidth allocation and management in a multitude of different communication mediums has generated some exceedingly tedious challenges that need to be addressed both intelligently and with innovative solutions. Current advances in high speed broadband technologies have manifold increased the amount of bandwidth required during successful multimedia streaming. The progressive growth of Neuro-Evolutionary techniques have presented themselves as worthy options to address many of the challenges faced during multimedia streaming. In this paper a Neuro-Evolutionary technique called the Recurrent Cartesian Genetic Programming Evolved Artificial Neural Network(RCGPANN) is presented for prediction of future frame sizes. The proposed technique takes into account the traffic size trend of the historically transmitted data for future frame size prediction. The predicted frame size forms the basis for estimation of the amount of bandwidth necessary for transmission of future frame. Different linear regression and probabilistic approaches are employed to estimate the allocated bandwidth, while utilizing the predicted frame size. Our proposed intelligent traffic size prediction along with bandwidth estimation and management results in a 98% increased efficiency.**

***Keywords*—scheduling, evolutionary algorithm, traffic estimation, MPEG-4, bandwidth allocation.**

I. Introduction

In the current scheme of major technological advancements, many new handheld devices such as smart phones and tablets have emerged having a vast improvement in term of their processing power and memory. These devices have the tendency to capture very high resolution images and videos. This high resolution content requires higher data rates while transference and streaming across the internet leading to some serious issues while appropriate bandwidth utilization. In a wireless network constrained by the available bandwidth, users tend to contest for channel availability time to support their respective multimedia flows. An efficient bandwidth allocator and scheduler becomes an essential entity for proper and optimum scheduling of multimedia traffic while multimedia streaming in such cases.

Recent times have seen a lot of effort being put into devising original and superior techniques to tackle the issue of efficient bandwidth scheduling and allocation. One of the popular means, among these methods include techniques based on Kalman Filtering. Bandwidth Available in Real Time (BART) proposed by Bergfeldt et. al [1] is one such technique. It uses self induced congestion in a packet switched network along with Kalman filtering for bandwidth estimation. An extension to BART has been proposed by Sedighizad et.al [2], known as Multi Rate BART (MR-BART), which tends to converge faster than its predecessor BART. Sometimes a combination of two or more techniques might lead to a novel and state of the art solution for bandwidth scheduling. Such combinations tend to merge the benefits of the contributing techniques. One such example lies in the combination of genetic algorithm, neural networks and fuzzy logic proposed by Chin-Teng et. al [3], known as Genetic Algorithm-based Neural Fuzzy Decision Tree (GANDFT). Probing based bandwidth estimations [4] [5] are also quite popular for proper bandwidth allocation and scheduling of variable bit rate (VBR) multimedia traffic. Such techniques make use of probe packets to collect information regarding channel utilization and available bandwidth and based on this information proper scheduling of the multimedia traffic is performed.

Some popular techniques employed while bandwidth estimation make use of frame size forecasters to predict the size of the future frames or packets. Bandwidth allocation is then performed based on the predicted size. An efficient size forecaster can thus play a vital role in the proficient estimation of required bandwidth for packet transmission, thus maintaining a fair amount of Quality of Service (QoS). Yi-Hsien et al. [6] have proposed Variable Step Size Normalized Least Mean Square Algorithm (VSSNLMS) for traffic prediction. Their proposed technique is quite efficient but tends to introduces a large error while predicting frames at the scene boundaries. Yuang and Tien [7] devised an intelligent bandwidth estimator based on an online traffic predictor utilizing the benefits of neural networks and fuzzy logic having two phase learning (i.e. Structural and parameter learning). For long term traffic prediction Lee and Chang [8] proposed an efficient ρ-domain rate model as a predictor. The technique when compared against Adaptive Least Mean Square(ALMS) [9] proposed by Yoo and Least Mean Square(LMS) algorithm [10] by Adas was shown to outperform both of these techniques.

The current research focuses on introducing a Neuro-Evolutionary Technique known as the Cartesian Genetic Programming Evolved Artificial Neural Network (CGPANN) [11] to develop an MPEG-4 frame size predictor, whose results are then employed for efficient bandwidth estimation. Neuro Evolution leads to the artificial evolution of all the components of an Artificial Neural Network (ANN) i.e. the evolution of network inputs, neuron inputs, weights, functions and network outputs as well as the complete network topology. It leads to new, innovative and efficient solutions as the entire ANN structure is altered during evolution. A CGPANN can either be feedforward or recurrent. A Recurrent CGPANN is employed to solve non-linear problems. The research proposed here also employs the Recurrent CGPANN (RCGPANN). The proposed solution is quite efficient in terms of problem solving during proper bandwidth allocation and management. Compared to an ANN, CGPANN gives simpler yet better solution in terms of implementation.

II. Cartesian Genetic Programming Evolved Artificial Neural Network (CGPANN)

Engulfing the core concept of Neuro-Evolution, the CGPANN makes use of Cartesian Genetic Programming (CGP) [12] to evolve an artificial neural network (ANN). A CGPANN involves complete component-wise evolution of the entire neural network leading to an innovative, robust and highly efficient solution [11]. The evolutionary process is performed continuously until an efficient system having the best fitness value is obtained. A CGPANN, like an ANN can also be feed-forward and recurrent (i.e. FCGPANN and RCGPANN). In the current research a Recurrent CGPANN (RCGPANN) based frame size predictor is evolved. The predicted frame size from the evolved network is then utilized for bandwidth allocation and estimation for different number of users under different constraints on the total available bandwidth. The next section gives details regarding the RCGPANN composition and evolution strategy.

III. Recurrent Cartesian Genetic Programming Evolved Artificial Neural Network (RCGPANN)

Based on the 'Jordan Network', an RCGPANN becomes the implementation choice, whenever the solution for a non-linear and dynamic problem is being devised. The neurons within the RCGPANN are identical in composition to its counterpart the FCGPANN. The only difference lies in the connection formed by these neuron inputs during evolution. Contrary to the FCGPANN, the neuron inputs of the RCGPANN may also form connections with outputs from recurrent nodes, in addition to the outputs of the preceding nodes or system inputs. A recurrent node takes in as input the system outputs and then multiplies these inputs with some weights within the range -1 to 1. These weighted values are then summed up and a standard activation function(log-sigmoid or tangent-hyperbolic) is applied on the sum to obtain the recurrent node output. A simple node and a recurrent node within the RCGPANN can be viewed from Fig. 1. In Fig. 1(a), the input R is basically the output taken from a recurrent node. It is not a compulsion for the simple node to form a connection with a recurrent node. The connection formed is basically on the basis evolution.

An RCGPANN can either be represented as a genotype (Fig. 2(a)) or as a phenotype(Fig. 2(b)). Each column in the genotype represents a simple node in the RCGPANN. The phenotype gives the evolved topological structure of the network specified by the genotype. The grayed column within the genotype and the dotted box within the phenotype of Fig. 2 are clear representations of nodes which are redundant or junk

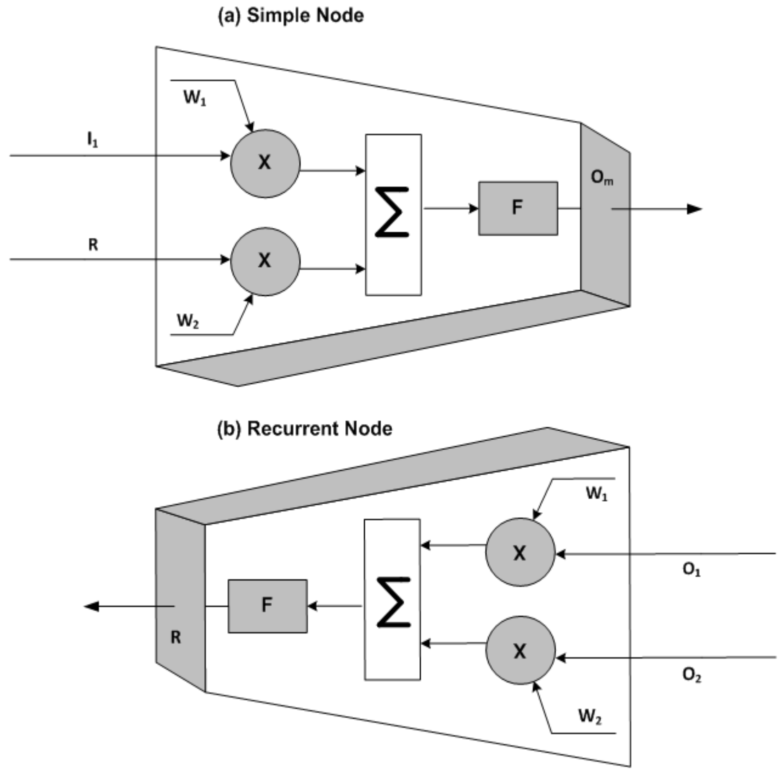

Fig. 1. Basic composition of (a) A Simple Node and (b) A Recurrent Node operating within an RCGPANN

nodes. This brings to light another quality of the RCGPANN which differentiates it from a typical recurrent ANN, i.e. output from every neuron within the layers of the RCGPANN is not being fed to the inputs of the subsequent nodes or network outputs. Such non-utilized neurons are termed as junk nodes. The junk nodes are produced during evolution along with the rest of the network components and topology. The selection of a node output, a recurrent node output or a system input as a connection to an RCGPANN node or network output is performed using a Pseudo Random Number Generator [13]. The evolution of RCGPANN follows certain steps of implementation detailed in the next section.

A. RCGPANN Algorithm

The technique involved in evolving an RCGPANN can be viewed from the algorithm given below.

1) **START**
   - **INPUTS**
     - N_{node}: Total number of neurons.
     - N_{input}: Total number of node inputs.
     - $S_{inputVector}$: The inputs to the network.
     - $Error_{fitness}$: Minimum error to check the fitness of the final genotype.
     - N_{gen}: Total number of genotypes in a generation.
     - $function_{act}$: The activation function list for the neurons.
     - N_{recurr}: Total number of recurrent nodes.

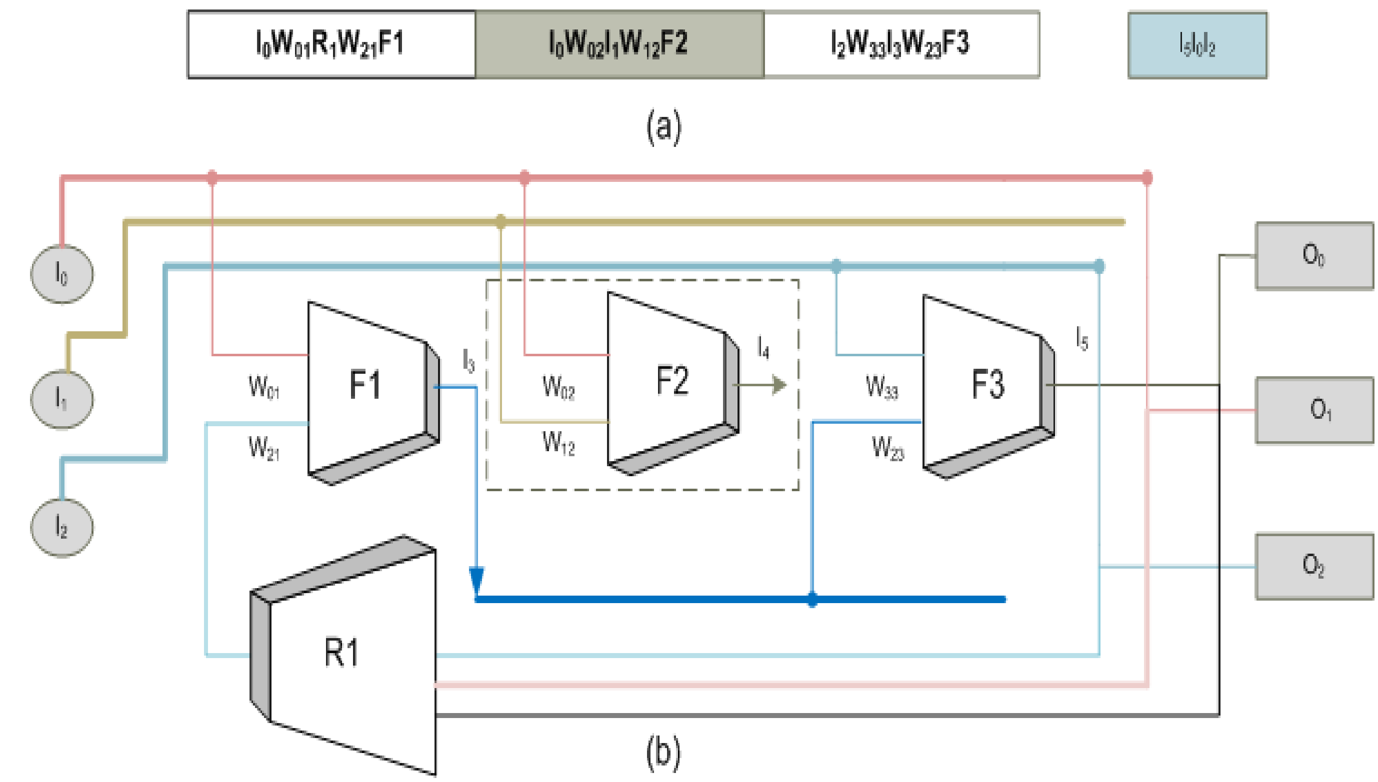

Fig. 2. (a) RCGPANN Genotype Representation and (b) RCGPANN Evolved Phenotype Representation

- $MR\%$: Mutation Rate.

- **OUTPUTS**
  - $S_{outputVector} = [op_1, op_2, ...op_M]$: Outputs from the network. Where M is the total number of system outputs

2) **INITIAL** Since no previous generations of the genotypes exist, therefore at the initial stage a population of N_{gen} genotypes is created. For this purpose the neurons or genes of every genotype are produced using Equation. 1, 2 and 3. Equation. 1 connects the node inputs to the outputs of preceding nodes, system inputs or recurrent inputs depending on a value generated by a random number generator [13]. Thus σ specifies the genotype number, ψ gives the node number and ϕ relates to the node input. Equation. 2 assigns a weight to each input of a node in the range [-1 to 1]. χ in Equation. 2 refers to the weight of the node input. This value is also assigned using a pseudo-random number generator which produces the index of the weight in the range to be assigned. Equation. 3 assigns a function to the neurons from the vector $function_{act}$. Activation functions are the same as the ones utilized in an ANN (i.e log-sigmoid, tangent-hyperbolic etc). $\psi.oper$ in equation 3 basically refers to the activation function assigned to the node. The index for the function in the $function_{act}$ list is again generated using a pseudo random number generator.

$$\begin{aligned} net_{genotype}(\sigma, \psi, \phi) = PRG([S_{inputVector} : \\ net_{genotype}(\sigma, \psi - 1), \ldots, net_{genotype}(\sigma, 1) : \\ net_{recurrent}(1), \ldots, net_{recurrent}(N_{recurr})]) \end{aligned} \tag{1}$$

$$net_{genotype}(\sigma, \psi, \chi) = PRG([-1, \ldots, 1]) \tag{2}$$

$$net_{genotype}(\sigma, \psi.oper) = PRG(function_{act}) \tag{3}$$

The genotype or network outputs are generated using Equation. 4. The outputs can either be connected with the system inputs or the node outputs. The connection to the output is also chosen based on the index generated by a pseudo random number generator. op_m in Equation. 4 is the m^{th} output of $S_{outputVector}(m = 1, ..., M)$.

$$\begin{aligned} op_m = PRG([S_{inputVector} : net_{genotype}(\sigma, \psi), \\ \ldots, net_{genotype}(\sigma, 1)] \end{aligned} \tag{4}$$

A recurrent node is also created using Equation. 2 and 3, which assign weights and functions respectively to the recurrent node.

3) **FITNESS EVALUATION and SELECTION** The fitness of all ten genotypes (parent and offspring) is calculated. Fitness is actually a measure of the Root Mean Square Error (RMSE) [14] produced by a genotype network. From the N_{gen} genotypes the fittest(i.e. the one having the least RMSE) is chosen

as the parent for mutation. If the RMSE of the parent is less than $error_{fitness}$, then evolution stops and the chosen parent genotype is selected as the final evolved network.
During fitness evaluation if a parent and the child have the same and highest fitness, then the child is chosen as the parent for next generation [12].

4) **MUTATION** The parent genotype is now mutated to produce $N_{gen} - 1$ offspring. Mutation is performed by changing either the simple nodes input (using Equation. 1), weights (equation 2), functions (Equation. 3) or system outputs (Equation 4). The recurrent nodes can also be mutated while producing new offspring. The mutation is performed by $MR\%$ (the mutation rate). Mutation is performed on the $1+\lambda$ strategy given by Miller and Thompson [12](i.e. λ offspring for a single parent). After mutation the N_{gen} genotypes (parents along with the offspring) are again sent to step 3 for fitness evaluation of the chromosomes.
5) **END**

IV. RCGPANN Frame Size Predictor Simulation Setup

For the current proposed RCGPANN packet size estimator, ten setups were evolved using the algorithm specified in Section. III-A having different number of total nodes(50, 100, 150, 200, 250, 300, 350, 400, 450, 500). These include the both the active and the junk nodes. The mutation rate for the networks is set at 10%, as this value has been shown to give exemplary results [11]. All the networks take in as input the frame sizes of ten past frames thus leading to networks having 10 system inputs. All the setups have 10 system outputs which are averaged to produce the predicted packet size. The number of maximum recurrent nodes for these setups has been set at 10. Thus the evolved systems have 10 feedbacks, of which some or all might not be utilized. The predicted frame size from the RCGPANN networks is then used further for bandwidth allocation to aide in lose less MPEG-4 traffic streaming. For the testing of the predictor a total of seven different MPEG-4 file sources have been taken. The list of the movies used during system testing can be viewed from Table. I. Mean Absolute Percentage Error (MAPE) [14] is used to evaluate the performance of the traffic predictor. The performance of the predictor is evaluated in detail in the next section (Section. V). The best packet predictor was obtained for 100 nodes and the experimental setup for the evolved system can be viewed from Fig. 3. From the figure it can be seen that after evolution, only five nodes are active in the end system. Furthermore only one recurrent input is being employed as the feedback input. It can also be observed that among the previous packet size values, the latest or the 10^{th} input plays the highest role in predicting the future packet size. It can also be seen that many of the system inputs (i_0, i_3, i_4, i_6) play no role in producing the future packet size. Mean Absolute Percentage Error (MAPE) is used as a criteria for judging the performance of the predictor.

The evolved packet size predictor is given in Equation. 5. The exponential expression(standard log-sigmoid function) in the equation gives the composition of the 13^th node and N_9 is the ninth node expressed in Equation 6. Equation. 7, 8 and 9 give the zeroth, second and first node which are inputs to the 9^th node. $i_0, i_1, ..., i_9$ are the system inputs.

TABLE I. Testing Data

S.No	MPEG4-Sources	Number of Frames	Average to Peak Ratio	MAPE (%) for 100 Nodes
1	First Contact	50,712	0.13	1.9065
2	Silence of the Lambs	50,287	0.09	1.2808
3	Star Wars IV	37,536	0.12	1.8566
4	The Firm 1	65,527	0.06	1.4132
5	The Firm 2	65,527	0.23	1.4338
6	From Dusk till Dawn	52,520	0.21	3.2529
7	Starship Troopers	65,529	0.21	1.8649

$$packetSize = \frac{1}{10}\{7i_9 + i_7 + i_8 + \frac{1}{1 + e^{3.9334N_9 - 0.9911i_5}}\} \tag{5}$$

$$N_9 = \frac{1}{1 + e^{1.9782N_2 + 0.9962i_5 + 0.9801N_0}} \tag{6}$$

$$N_0 = \frac{1}{1 + e^{1.996i_2 + 2.9103i_5}} \tag{7}$$

$$N_2 = \frac{1}{1 + e^{1.9565i_5 + 1.9704i_9 - 0.9775N_1}} \tag{8}$$

$$N_1 = \frac{1}{1 + e^{-1.296i_5 + 0.2794R_0 - 0.4212i_1}} \tag{9}$$

Different probabilistic and linear regression methods are utilized to further estimate the bandwidth required for frame transmission using the frame size estimated by the best 100 node RCGPANN setup. The results for frame size prediction will be presented and evaluated in the next section.

V. Traffic Prediction Results and Analysis

Table. I shows that for the frame predictor evaluation, more than 80,000 frames of 7 different movies were taken into consideration. These movies were selected randomly and gave rise to a considerably huge data set. The results are indicative of the efficiency of the frame predictor. MAPE (%) was calculated for the estimated size values against the actual values. It can be observed from the table that a minimum MAPE of 1.2% has been achieved, thus proving the accuracy with which the predictor forecasts the size values.

Fig. 4 shows the MAPE for simulation setups with different number of nodes. The figure indicates that the setup with 100 nodes gives the best solution. The traffic predictor proposed in the current research has been compared against some very popular techniques. Observing Table. II, the comparison greatly supports the use of the proposed predictor as opposed to its predecessors.

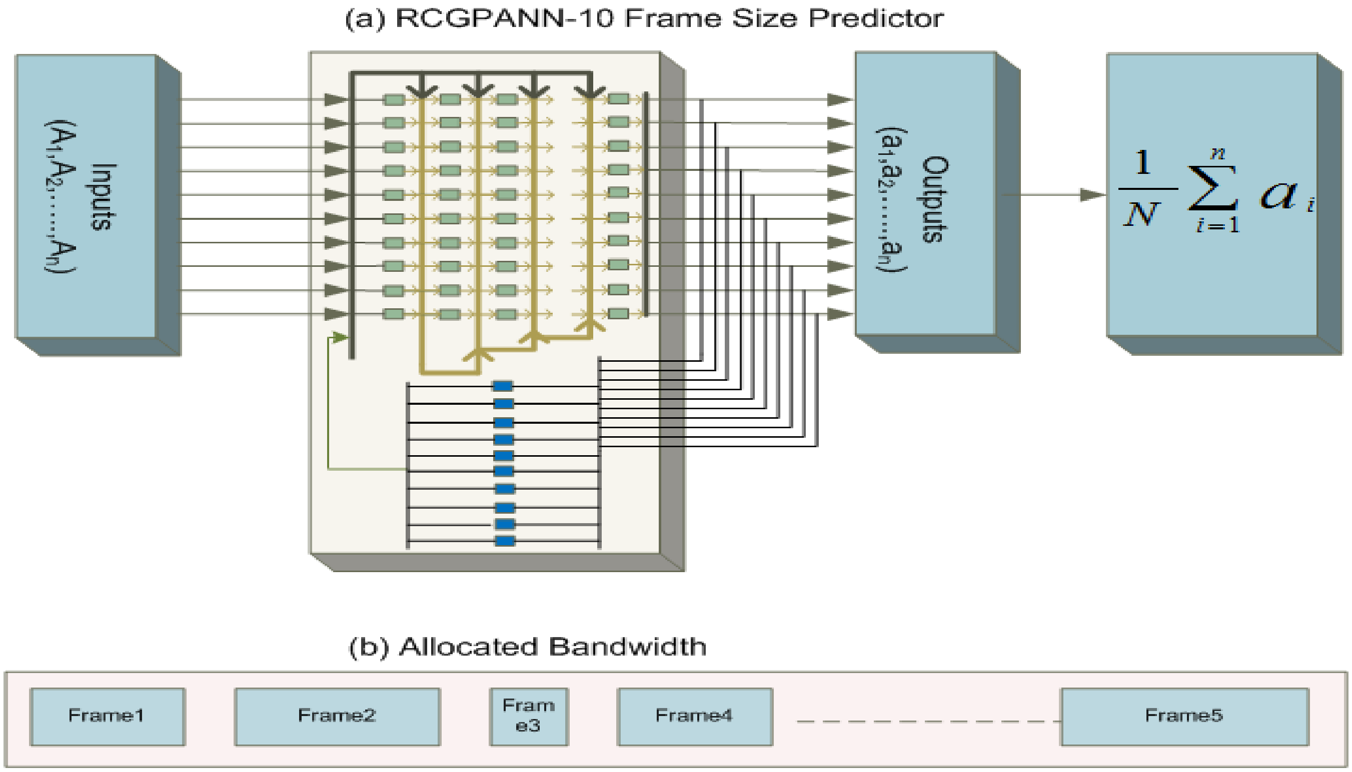

Fig. 3. (a) Evolved Packet Size Estimator (b) Allocated Bandwidth

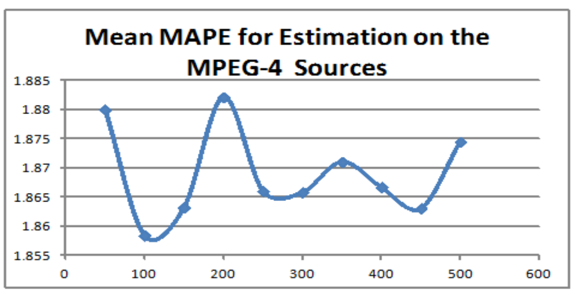

Fig. 4. Mean MAPE for Systems with Different Number of Nodes

TABLE II. FRAME PREDICTION COMPARISON

S.No	Technique	% Prediction Error
1	LMS [10]	10.7
2	ALMS [9]	7.4
3	$\rho - LSP$ [8]	5.2
4	SARIMA [15]	1.37
5	[16]	1.4
7	**Proposed Scheme RCGPANN**	**1.2**

VI. BANDWIDTH ALLOCATION SIMULATION SETUP

To compare bandwidth allocation efficiency as a consequence size prediction by the RCGPANN predictor, different simulation scenarios have been taken under consideration. These scenarios differ from each other based on the number of users streaming the MPEG-4 data, the percentage bandwidth available and the different limitations under which bandwidth allocation is performed. Basic categorization of the simulation scenarios involves four types of bandwidth allocation setups. The first setup involves the fixed data rate scenario in which 10% of the users from the total number of users are streaming traffic based on a fixed packet size. Fixing the packet size results in allocation of a fixed bandwidth to these users. For the rest of the users, the RCGPANN predicts the packet size and bandwidth allocation is performed according to the predicted size.

The second scenario resembles the fixed data rate scenario in the sense that in this case, 10% of the users are being given a higher priority for data streaming as compared to the other beneficiaries of this scheme. Bandwidth allocation has now been prioritized. Fig. 5 shows the setup involved for fixed data rate or priority based bandwidth allocation.

The variable data rate case is the simplest where the required bandwidth is simply estimated and allocated for frame based on the predicted frame size.

The probability based bandwidth allocation setup however allocates bandwidth based on numbers generated by pseudo random number generator. Basically bandwidth is calculated for each user based on the predicted packet size. Using the estimated value, a number is assigned to the user from a predefined range. A random number is then generated lying in this predefined range. If the number assigned to the user is close to but greater than the generated random number, then the packet for the user is dropped, otherwise it is transmitted. Fig. 6 shows the setup for probability based bandwidth allocation.

VII. BANDWIDTH ALLOCATION RESULTS AND ANALYSIS

The bandwidth allocation efficiency has been calculated in terms of packet loss for different number of users, while

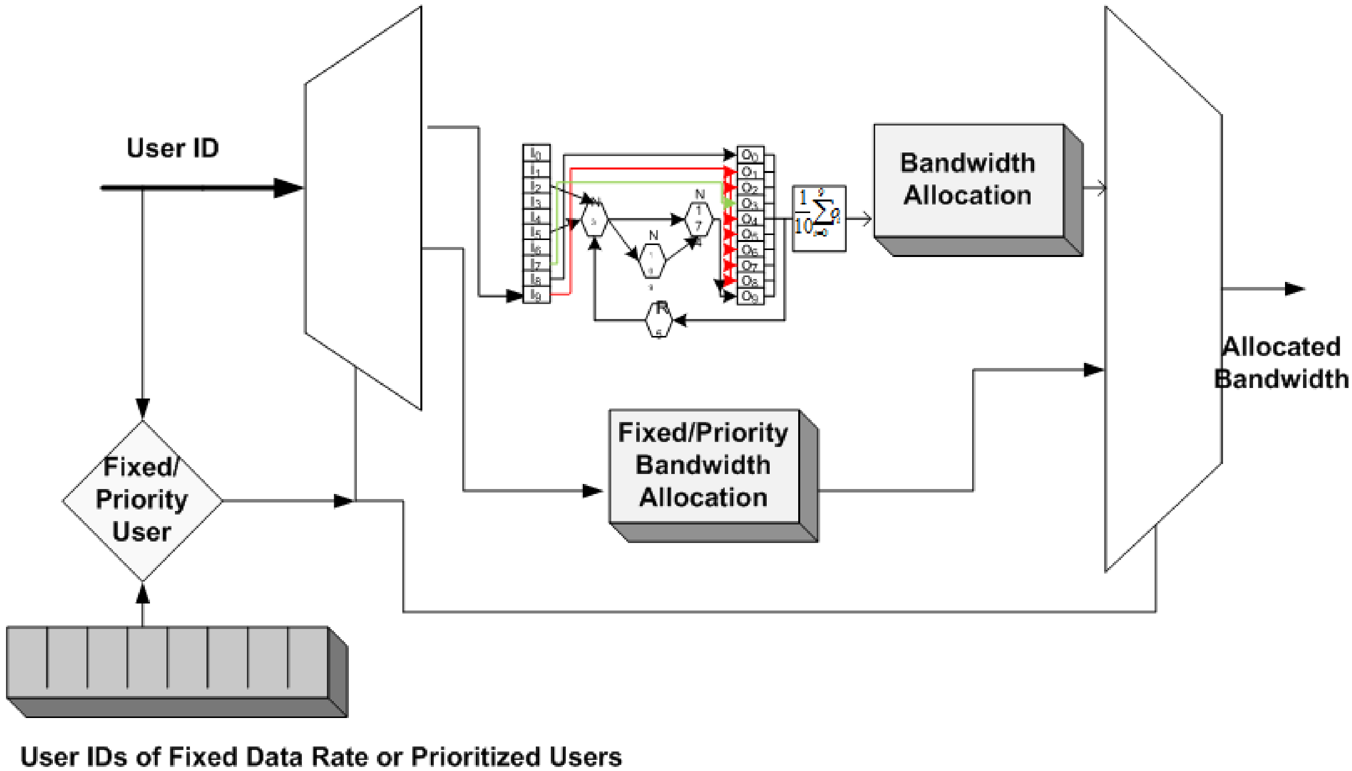

Fig. 5. Fixed/Priority based Bandwidth Allocator

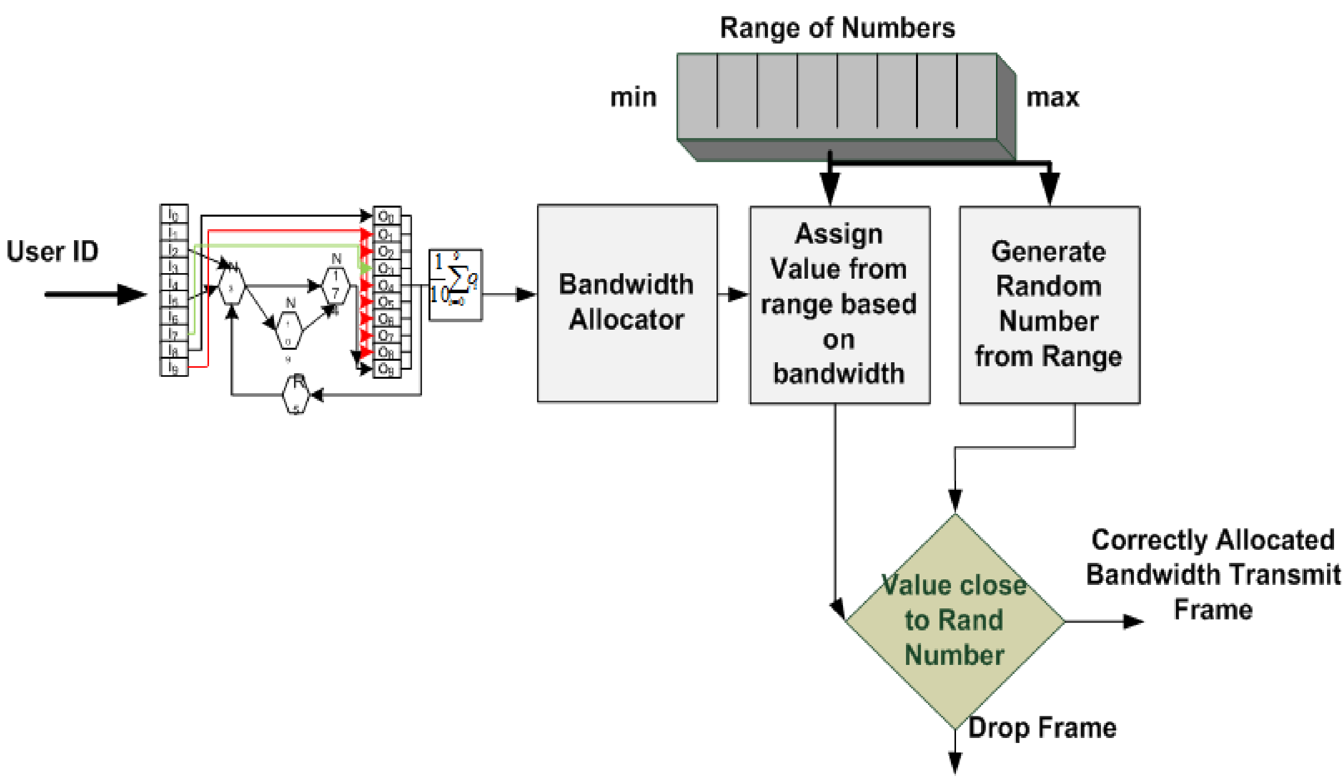

Fig. 6. Probability based Bandwidth Allocator

employing the simulation setups detailed in Section. VI. Table. III gives details regarding the packet loss resulting from bandwidth overhead. Bandwidth overhead occurs when the actual required bandwidth exceeds the estimated amount of bandwidth. From Table. III, it can be seen that using the proposed bandwidth allocation scheme considerable packet

TABLE III. AVERAGE FRAME DROP DUE TO BANDWIDTH OVERHEAD

Users	Scenarios	% Bandwidth Allocation									
		15%		25%		35%		50%		100%	
		Act	Est	Act	Est	Act	Est	Act	Est	Act	Est
10	Fixed	522	523	333	289	96	96	1	9	0	0
	Variable	496	494	194	170	38	23	9	0	0	0
	Probability	453	272	135	112	26	11	0	0	0	0
	Priority	464	462	195	169	49	30	1	0	0	0
50	Fixed	594	409	65	57	2	0	0	0	0	0
	Variable	429	429	76	69	1	0	0	0	0	0
	Probability	393	311	55	35	1	0	0	0	0	0
	Priority	429	429	71	63	2	0	0	0	0	0
100	Fixed	595	419	49	46	0	0	0	0	0	0
	Variable	414	414	60	56	0	0	0	0	0	0
	Probability	367	318	35	24	0	0	0	0	0	0
	Priority	417	417	48	45	0	0	0	0	0	0
500	Fixed	416	416	28	27	0	0	0	0	0	0
	Variable	418	418	25	25	0	0	0	0	0	0
	Probability	369	312	18	17	0	0	0	0	0	0
	Priority	419	419	28	27	0	0	0	0	0	0
1000	Fixed	420	420	28	14	0	0	0	0	0	0
	Variable	423	423	17	16	0	0	0	0	0	0
	Probability	375	308	15	10	0	0	0	0	0	0
	Priority	417	417	15	14	0	0	0	0	0	0

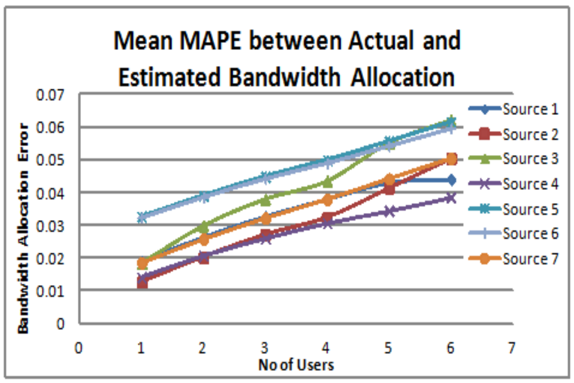

Fig. 7. Bandwidth Allocation Error for 7 MPEG-4 Sources(Table I for different users

loss reduction occurs. When the waiting time for a packet reaches its expiry time, then this results in dropping of the packet. Packets are dropped as a result of overhead because they do not get the required bandwidth for transmission and reach their expiry time while waiting in the queue to be transmitted.

Fig. 7, shows the actual versus estimated bandwidth allocation error for 6 users. The error has been calculated the seven different sources given in Table. I. From the figure it can be seen that the bandwidth allocation efficiency of the proposed technique gives more than 98% efficient results.

In Table. IV, the bandwidth estimation efficiency of our proposed technique has been compared against other techniques provided in literature. The first is an intelligent bandwidth estimation technique proposed by Yuan et.al [17] which attempts at dealing only with multimedia packets available at the application layer. The other technique is a probing based bandwidth estimation technique proposed by Hu and Steenkiste [5]. From the table it can be seen that the proposed technique outperforms both of its contemporaries.

VIII. CONCLUSION

The paper presents a novel packet size estimator utilizing the Neuro-Evoltionary technique RCGPANN. Based on the

TABLE IV. BANDWIDTH ESTIMATION EFFICIENCY FOR 3 USERS

S.No	Technique	% Error Rate(Actual Vs Estimated)
1	iBE [17]	0.14
2	Spruce [5]	1.19
3	Proposed Scheme with FCGPANN	0.0128

predicted size, efficient bandwidth management and allocation is performed. The evolved RCGPANN setup gives an efficient solution composed of only 5 active nodes from a total of 100 nodes and a frame prediction efficiency of approximately 99%. Based on the predicted frame size, the required bandwidth for frame transmission is calculated using different probabilistic and linear regression approaches. The bandwidth allocation and estimation efficiency has been evaluated under different scenarios and the proposed technique gives a bandwidth allocation efficiency of more than 98%. To conclude the method proposed here is quite efficient in terms of its implementation and purpose.

REFERENCES

[1] E. Bergfeldt, S. Ekelin, and J. M. Karlsson, “Real-time available-bandwidth estimation using filtering and change detection,” *Comput. Netw.*, vol. 53, no. 15, pp. 2617–2645, Oct. 2009.

[2] M. Sedighizad, B. Seyfe, and K. Navaie, “Mr-bart: Multi-rate available bandwidth estimation in real-time,” *J. Netw. Comput. Appl.*, vol. 35, no. 2, pp. 731–742, Mar. 2012.

[3] C.-T. Lin, I.-F. Chung, H.-C. Pu, T.-H. Lee, and J.-Y. Chang, “Genetic algorithm-based neural fuzzy decision tree for mixed scheduling in atm networks,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 32, no. 6, pp. 832–845, 2002.

[4] S. Y. Nam, S. Kim, J. Kim, and D. K. Sung, “Probing-based estimation of end-to-end available bandwidth,” *Communications Letters, IEEE*, vol. 8, no. 6, pp. 400–402, June 2004.

[5] N. Hu and P. Steenkiste, “Evaluation and characterization of available bandwidth probing techniques,” *IEEE Journal on Selected Areas in Communication*, vol. 21, no. 6, pp. 879–894, Sep. 2006.

[6] Y.-H. Tseng, E.-K. Wu, and G.-H. Chen, “Scene-change aware dynamic bandwidth allocation for real-time vbr video transmission over ieee 802.15.3 wireless home networks,” *Multimedia, IEEE Transactions on*, vol. 9, no. 3, pp. 642–654, april 2007.

[7] M. C. Yuang and P. L. Tien, “Multiple access control with intelligent bandwidth allocation for wireless atm networks,” *IEEE Journal on Selected Areas in Communication*, vol. 18, no. 9, Sep. 2006.

[8] C.-C. Lee and P.-C. Chang, “A novel long-span traffic predictor for real-time vbr videos via ρ-domain rate model,” *Communications Letters, IEEE*, vol. 9, no. 3, pp. 279–281, 2005.

[9] S.-J. Yoo, “Efficient traffic prediction scheme for real-time vbr mpeg video transmission over high-speed networks,” *Broadcasting, IEEE Transactions on*, vol. 48, no. 1, pp. 10–18, 2002.

[10] A. Adas, “Supporting real time vbr video using dynamic reservation based on linear prediction,” in *INFOCOM ’96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE*, vol. 3, mar 1996, pp. 1476 –1483 vol.3.

[11] M. M. Khan and G. M. Khan, “A novel neuroevolutionary algorithm: Cartesian genetic programming evolved artificial neural network (cg-pann),” in *Proceedings of the 8th International Conference on Frontiers of Information Technology*, ser. FIT ’10, 2010, pp. 48:1–48:4.

[12] J. F. Miller and P. Thomson, “Cartesian genetic programming,” vol. 1802, 2000, pp. 121–132.

[13] N. Nisan, “Pseudorandom generators for space-bounded computation,” *Combinatorica*, vol. 12, no. 4, pp. 449–461, 1992.

[14] J. Armstrong and F. Collopy, “Error measures for generalizing about forecasting methods: Empirical comparisons,” *International Journal of Forecasting*, vol. 8, no. 1, pp. 69–80, 1992.

[15] G. Trlin, "Vbr video frame size prediction using seasonal arima models," in *Software, Telecommunications and Computer Networks (SoftCOM), 2012 20th International Conference on*, 2012, pp. 1–5.

[16] M. G. Jibukumar, R. Datta, and P. Biswas, "Kalman filter based variable bit rate video frame size prediction," in *Wireless Pervasive Computing, 2008. ISWPC 2008. 3rd International Symposium on*, 2008, pp. 459–463.

[17] Z. Yuan, H. Venkataraman, and G.-M. Muntean, "ibe: A novel bandwidth estimation algorithm for multimedia services over ieee 802.11 wireless networks," *Wired-Wireless Multimedia Networks and Services Management*, pp. 69–80, 2009.