Neural Network Based of a New User IP Address Determination and Address Conflict Elimination

Raid W. Daoud*, Wissam S. Hassan
Department of Electronic Techniques, Al-Hawija Tech. Inst., Northern Technical University, Kirkuk, Iraq
raid.daoud@gmail.com

Abstract

In this paper, the neural network (NN) is used to control the remote IP address that related to user device. When the IP address assigned carefully and controlled by a novel tool, the work performance and service quality will be better. The input for the NN determined depending on the repeated problem in recent networks and the available parameters which are in the main server node. The training process done by determining the required training function and activation function for all neurons in the NN. The performance of the proposed method was ($1*10^{-20}$) or less in more cases and the error of the learning process is nearly ($1*10^{-6}$). In addition to IP address control, the proposed method is a good manner to optimize the security issue by specializing an IP address for a given user that can't be used by other device. Finally, the NN subject the produced IP address for the given users for test and validation which reached to the goal at smallest time (~0.0001s) and little number of iteration (9 - 20) epoch.

Keywords: IP address, Conflict IP, Network Communication and Neural Networks.

Paper History: (Received: 25/10/2017; Accepted: 01/04/2018)

Introduction:

In a communication network information is transferred from one node to another as data packets [1]. Successful operation of data communication network is critically dependent on the provision of an adequate routing algorithm. Routing algorithms are methods for finding the best way from a node to another node or user end [2][3]. Modern information society has the increasing need for Internet access, as the global network for information circulation. Nowadays, it is necessary to obtain Internet access for all types of network devices (fixed and mobile wireless devices), in any possible way and at any place. This is extremely important if we bear in mind that the use of wireless devices, like smart phones, mobile phones, PDAs, laptops etc. is on the increase. These demands impose the need for large infrastructure, which is often impossible to realize, especially in rural areas [4]. An important part of communication and information networks is active network elements that allow you to route data from one place to another. Some services have a higher priority than others, in particular services in real time [5]. Another important point of development is, for example, the increasing routing tables according to which the routers determine the path for each link [6]. Unpredictable and generally undesirable things tend to happen when multiple devices attempt to use the same IP address [7]. An address conflict scenario might go like this device A starts up and uses IP address 10.88.80.10 Other devices establish and carry on communications with Device A. Device B has incorrectly been configured to also use 10.88.80.10 Device B starts up Existing communications to Device A may be disrupted and re-established with Device B (though not necessarily). New communications initiated by other devices could go to Device A or Device B (or may be established and disrupted again).In such situations, it can be difficult to determine that there even is an address conflict situation, and where the offending device is located. To further complicate the problem, there is no universally implemented mechanism for detecting IP address conflicts [8]. Nowadays many type of devices need for connection to internet and in many type of places. Main goals of the link are increase the capacity and decrease the bit error rate [9][10]. see Figure 1.

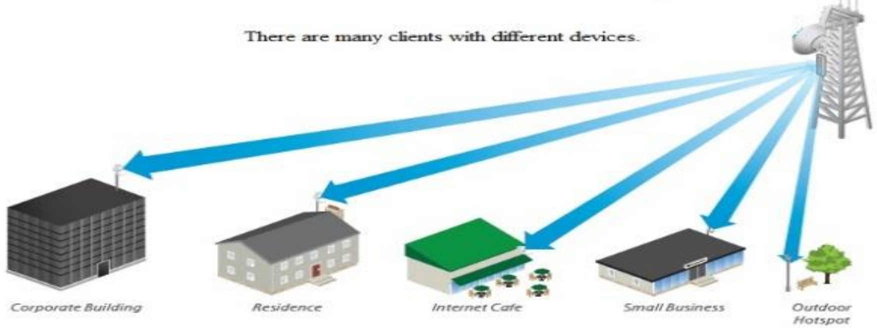

Figure 1: The types of devices that need the internet connection

In this proposed work, class A IP address is considered to give advance power for the network because it differ from those in standard network devices, class C.

Back-Propagation NN

The Back-Propagation NN (BPNN) is the most common network that can be used to learn the hard works. In this paper the BPNN used because there are no fixed equations or default output pattern to reach the goal. Instead of mathematical equations the BPNN tack place, that is receive the input parameters and produce the corresponding output depending on the target pattern. The network, which is used in this work, consists of three layers (input, hidden and output)[11], see Figure 2

.

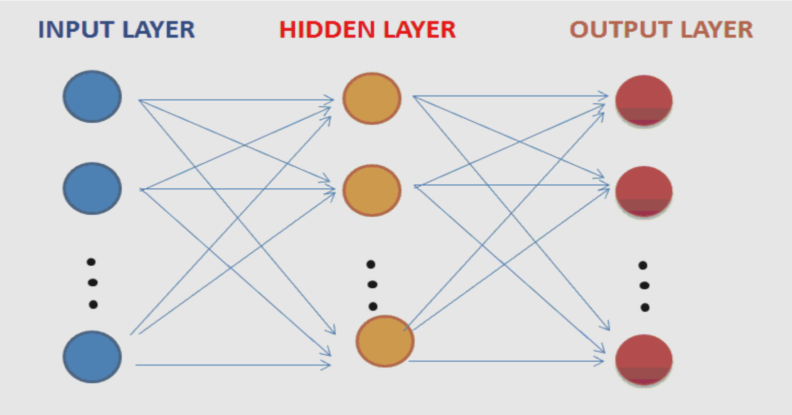

Figure 2: The structure of the BPNN used

The practical model for the BPNN which is used in this paper has 6-neurons in the input layer that hold the prepared 6-inputs variables. The inputs are determined carefully by studying the recent problems and use the available data in main node server. The activation function in the input and output is the 'pure-line' and in the hidden layer is 'tansig'. The hidden layer consist of 10-neurons where chosen to increase the accuracy of work and less operation time. The number of hidden layers neuron reached to 10 after many trial steps by increasing the neuron this form.

Input Parameters

The input parameters for the proposed network are chosen in a novel manner that can be used in the network optimization now and in other future work. The input parameters are combined between those in sender side and others in the receiver side to reach the full control. The list of the prepared inputs shown below:

1. Sector Number (device used for signal sending)
2. Type of Service (The server can supply different type of services)
3. Mean of Packets send/receive (Packets used to handshaking)
4. Signal Strength (Quality of signal in dB)
5. Receiver Type (More than one type of signal receiver type)
6. DHCP IP address (Dynamic Host Configuration Protocol)

The sector is a transmitter device in the main node, and each node has 3-sectors at least. The location of the user can be determined easily by knowing the sector number, because the first sector fixed to north direction and the second one rotate 120 ° toward left direction and the third one rotate 120° toward left of the second one. In addition to sectors, the 'Omni' is a transmitter device that covers 360° surround the node. Types of service or the bandwidth specified for the user, another parameter serve the proposed work to give the proper priority for the high service. The packets, which are the remote end (user device) repeatedly, send and receive them to the main server for handshaking process. The mean packets send/receive is addition priority competitive point. The signal strength and receiver type can be used to optimize its service according to specific given variables such as change the sector direction or by changing the power of the carrier signal. The last input variable, DHCP IP address, which is the IP address that the main node router supply it to the clients. The DHCP IP address can serve the work because it is impossible to repeat the same IP address for more than one client.

In recent days, the routers are not fully intelligent and in some cases connect the new client using the default IP address of the device. The default IP address is standard for some types of the receiver units. The conflict problem will appears if one case occurred of same default IP address copied to DHCP IP address. The servers go to restart or shutdown in some cases of IP conflict problem. The proposed system will solve this problem and other ones (user name attack) which will be discussed later.

Input Normalization

The NN is a package that receives only numbers to learn its neuron about specific form of output. The proposed system contains different types of inputs, such as: the device name, signal strength and sector number). See the Table-1, which summarizes all inputs and their kinds. The entire input variable must be in numbering form to begin in the process part. The normalization of the input variables is very important for the NN because its treatment with the numbers only. All i/p variables are translated to numbers between 0 and 1, according to their location in the Table-1. Each input vector divided into simple range of numbers that represent it for the NN. The DHCP IP address is entered for the NN as it's, with a scale to satisfy the normalization, because its importance for the un-repeated produced IP. The last two digits of the IP address will be the 6th input for the proposed NN, see Table 2. The Tx/Rx transferred to numbers 0.2, 0.25, 0.3 and 0.35 to represent the real values of the mean transmit and receive packets. The ratio of Tx/Rx represents the load of the link that is established between the node and a given user device.

Table 1: List for names of most efficient Input variables

i/p value i/p name	1'st	2'nd	3'rd	4'th
Sector No.	Sector-1	Sector-2	Sector-3	Omni
Type of Service	Default	Light	Fast	Extra
Mean Packets Tx/Rx	2000 bps	2500 bps	3000 bps	3500 bps
Signal Strength	-40 dB	-50 dB	-60 dB	-70 dB
Receiver Type	NanoBridg	NanoStation	NanoBeam	AirGrid
DHCP IP address	10.50.10.00	10.50.10.02	10.50.10.10	10.50.10.15

Table 2: The Normalized input.

i/p value i/p name	1'st	2'nd	3'rd	4'th
Sector No.	0.1	0.2	0.3	0.4
Type of Service	0.1	0.2	0.3	0.4
Mean Packets Tx/Rx	0.2	0.25	0.3	0.35
Signal Strength	0.4	0.5	0.6	0.7
Receiver Type	0.1	0.2	0.3	0.4
DHCP IP address	0.0	0.02	0.10	0.15

Training Procedure

The MATLAB R2012a is used to design of the proposed NN. Because the different inputs values and multiple output lines the Back-Propagation NN is used, in addition to more accurate result required. The proposed network contains three layers: Input, Hidden and Output layer. There are 6-neurons in the input layer to hold the 6-input variables. The hidden layer consists of 10-neurons and the output has 4-neurons which determine the IP address for the remote end (user device), see Table 3. The activation function for the input and output layers was ‘pure-line’, and the ‘tansig’ function was for the hidden layer.

Table 3: The output form of NN which represent the new IP address for the user device

Output 1	Output 2	Output 3	Output 4
10	00	00	00

The complex mathematical operations in the hidden layer required a graded function to satisfy the overall work. The final number, which will be produced from the neural, is the applied IP address for the client therefor the ‘pure-line’ is very compatible for this layer. The network learned for only 20-sample of inputs with pre-installed node variables. Table 4 summarized the data of the selected 20-inputs.

Table 4: Sample of 20-input for the NN

User no.	Sector	Signal	Receiver	Service	Packets	DHCP IP
1	0.1	0.4	0.2	0.2	0.3	0.1
2	0.1	0.5	0.1	0.2	0.2	0.2
3	0.1	0.5	0.2	0.2	0.35	0.3
4	0.1	0.7	0.2	0.1	0.35	0.4

5	0.1	0.4	0.3	0.1	0.3	0.5
6	0.2	0.4	0.3	0.1	0.3	0.6
7	0.2	0.6	0.4	0.2	0.25	0.7
8	0.2	0.5	0.2	0.3	0.3	0.8
9	0.2	0.4	0.2	0.2	0.3	0.9
10	0.2	0.7	0.2	0.1	0.3	0.10
11	0.3	0.4	0.4	0.3	0.2	0.11
12	0.3	0.5	0.3	0.2	0.2	0.12
13	0.3	0.5	0.3	0.2	3.5	0.13
14	0.3	0.4	0.4	0.1	0.3	0.14
15	0.3	0.6	0.4	0.1	2.5	0.15
16	0.4	0.6	0.1	0.1	2.5	0.16
17	0.4	0.5	0.2	0.3	0.3	0.17
18	0.4	0.4	0.2	0.3	0.3	0.18
19	0.4	0.4	0.1	0.2	0.3	0.19
20	0.4	0.4	0.1	0.3	0.35	0.20

The target must be fixed for the BPNN to learn the network and test the error. Table 5 shows the target numbers of the proposed network. The IP class A is used in this work, as mentioned early, for its specialty. Increment by 2 for each new IP was done in this work for more precision in the division process

Table 5: The target numbers for the BPNN

IP address			
0.10	0	0	0
0.10	0	0	0.2
0.10	0	0	0.4
0.10	0	0	0.6
0.10	0	0	0.8
0.10	0	0	0.10
0.10	0	0	0.12
0.10	0	0	0.14
0.10	0	0	0.16
0.10	0	0	0.18
0.10	0	0	0.20
0.10	0	0	0.22
0.10	0	0	0.24
0.10	0	0	0.26
0.10	0	0	0.28
0.10	0	0	0.30
0.10	0	0	0.32
0.10	0	0	0.34

0.10	0	0	0.36
0.10	0	0	0.38

Results

More than one parameter must be records when the NN is used. The outputs of the proposed network that related to each input parameters listed in the Table 6. The produced IP addresses will supply for the end user device to solve the mentioned problems. In addition to the output, the weights and bias for each neuron must be back up for future new input variable.

The error is very small by comparing the output of the BPNN with the target; see Table 7, the mean error ration nearly (1*10-4). In some cases the error jump to a low ratio (1*10-2) but this field is less significant values, because it is a network ID that may be little effect on the IP. The last column of the output is of very important, showing in Table 6, values which represent the identification number for the user device. The end used device will take this IP address and not repeated in any other ones.

The BPNN in real components is shown in Figure 3, which contains the number of inputs and hidden layers in addition to output with their activation function. The training curve, which shows the network performance and the smooth transition, in addition to test and validation curves are shown in Figures 4 and 5, for a given input variable (user no. 4). The number of iteration is differing from input to other; see Figures 4 and 5, depending on the error ratio which was reached. There are 12 epochs in the 4th input variable and 9 epochs for the 5th input variable. In some cases, the iteration number reaches to maximum allowed number if there is no proper solution. The Figure 4 shows a best validation and test performance for the assigned group of the input patterns, with approximately $1*10^{-20}$ validate ratio. The training and validation are the response of the NN which reflect the well design and operation of the proposed system; as shown, the two curves are sketched carefully.

Table 6: The produced IP addresses from the NN for the given 20-sample

IP address			
0.10008	$-0.0*10^{-5}$	0.0518	$0.0112*10^{-2}$
0.10005	$0.0*10^{-3}$	$0.0*10^{-6}$	$20.042*10^{-2}$
0.08356	$-0.0*10^{-7}$	$0.0*10^{-5}$	$40.012*10^{-2}$
0.10008	$0.0*10^{-7}$	$0.0*10^{-6}$	$60.012*10^{-2}$
0.10008	$0.0*10^{-6}$	$0.0*10^{-6}$	$80.012*10^{-2}$
0.08845	$0.0*10^{-6}$	$0.0*10^{-7}$	$99.928*10^{-2}$
0.10009	$0.0*10^{-5}$	$0.0*10^{-7}$	$12.0001*10^{-2}$
0.10004	$-0.0*10^{-5}$	$0.0*10^{-7}$	$14.0001*10^{-2}$
0.10004	$0.0*10^{-3}$	$0.0*10^{-4}$	$15.8995*10^{-2}$
0.10004	$0.0*10^{-5}$	$0.0*10^{-3}$	$18.0001*10^{-2}$
0.10008	$0.0*10^{-6}$	$0.0*10^{-3}$	$20.0001*10^{-2}$
0.10005	$0.0*10^{-6}$	$0.0*10^{-3}$	$22.0001*10^{-2}$
0.10003	$0.0*10^{-6}$	$0.0*10^{-7}$	$24.0001*10^{-2}$
0.09812	$0.0*10^{-5}$	$0.0*10^{-6}$	$25.9985*10^{-2}$
0.10008	$0.0*10^{-5}$	$0.0*10^{-7}$	$28.0001*10^{-2}$
0.10008	$0.0*10^{-7}$	$0.0*10^{-7}$	$30.0001*10^{-2}$
0.10004	$0.0*10^{-7}$	$0.0*10^{-7}$	$32.0001*10^{-2}$
0.10001	$0.0*10^{-7}$	$0.0*10^{-5}$	$34.0001*10^{-2}$
0.10001	$0.0*10^{-7}$	$0.0*10^{-6}$	$36.0021*10^{-2}$
0.10008	0.0021	$0.0*10^{-6}$	$38.0001*10^{-2}$

Table 7: The errors for the 20-samples input variable

Error1	Error2	Error3	Error4
0.000080	0.000010	0.051800	0.000100
0.000050	0.001000	0.000001	0.000400
-0.016440	0.000000	0.000001	0.000100
0.000080	0.000000	0.000001	0.000100
0.000080	0.000001	0.000001	0.000100
-0.011550	0.000001	0.000000	0.899200
0.000090	0.000001	0.000000	0.000010
0.000040	0.000010	0.000000	0.000001
0.000040	0.000010	0.000100	-0.001005
0.000040	0.000001	0.001000	0.000001
0.000080	0.000001	0.001000	0.000001
0.000050	0.000001	0.001000	0.000001
0.000030	0.000001	0.000000	0.000001
-0.001880	0.000001	0.000010	-0.000015
0.000080	0.000010	0.000000	0.000001
0.000080	0.000000	0.000000	0.000001
0.000040	0.000000	0.000000	0.000001
0.000010	0.000000	0.000001	0.000001
0.000010	0.000000	0.000000	0.000021
0.000080	0.002100	0.000000	0.000001

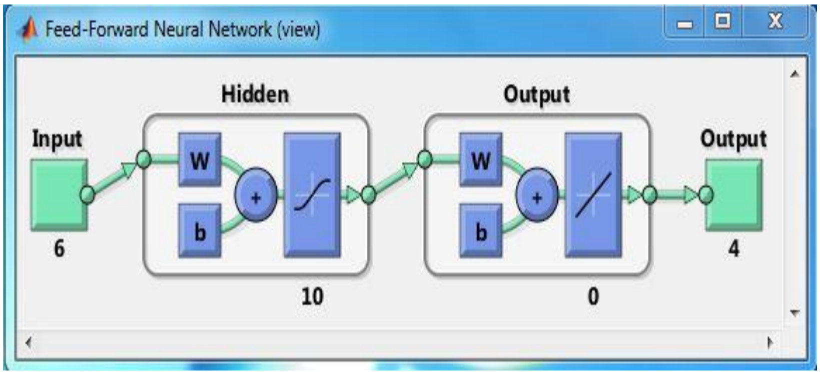

Figure 3: The BPNN components with all layers.

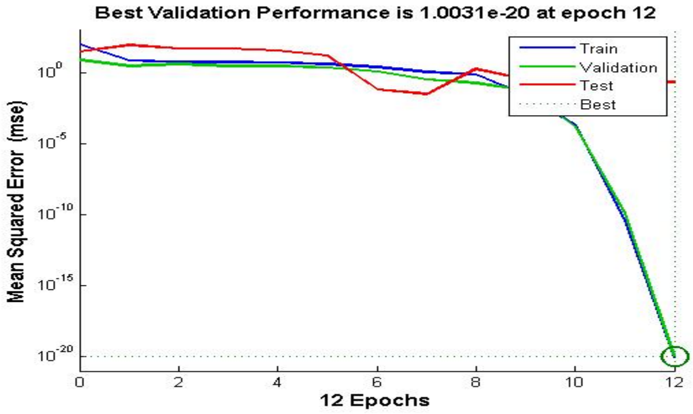

Figure 4: The Performance of the BPNN and its validation curve for 4'th input.

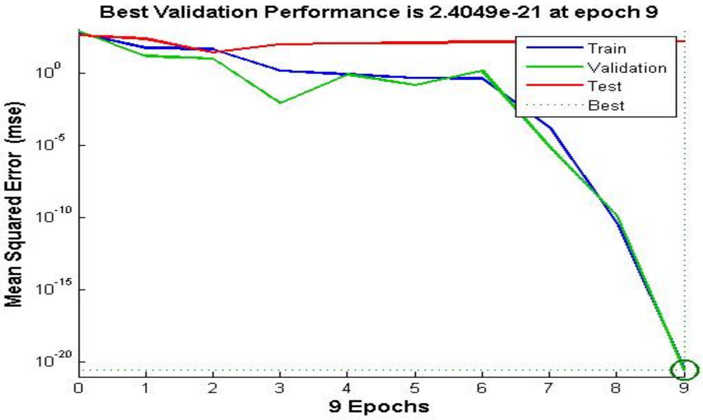

Figure 5: The Performance of the BPNN and its validation curve for 5'th input.

Conclusion

The IP control for the end user device is proposed so that the remote device IP addresses assigning gives the node server more stability and no IP conflict signal occurrence. By using the NN, the system was optimized and became more realized for its capability. The hidden layer well designed and activation function selected to achieve its function. The produced IP address has a very small error ratio which gives the proposed method more acceptances.

References

[1] A. N Khloud., Y. A Turki. and Y. A Abdulkareem. Computer Network Routing Using Fuzzy Neural Networks, Basrah Journal of Science, 31, 2 (2013) 20-35.

[2] J. Malrand, Communication Networks: A First Course, R. D. Irwin and Akson Associates, Inc, 1991.

[3] W. Newton, A Neural Network Algorithm for Internetwork Routing, Report in Software Engineering, for Degree of Bachelor, 2002.

[4] K. Nenad, R. Irini and R. Branimir, A Neural Networks-Based Hybrid Routing Protocol for Wireless Mesh Networks, Sensors, 12, 7548-7575, doi: 10.3390/s120607548, 2012.

[5] S. Vladislav and P. Roman, Training a Neural Network for a New Node Element Design, ISSN 0033-2097, R. 89 NR 2b, 2013.

[6] A. Takizava and A. Fukasawa, Novel neural Network Scheme Composed of Prediction and Studies. Proceedings of the 13'Th WSEAS International Conference on Systems, pp. 611-615, 2009, ISBN: 978-960- 474-097-0, ISSN: 1790-2769.

[7] S. Kent, R. Atkinson, Security Architecture for the Internet Protocol, RFC-2401, IETF, 2003.

[8] M. Kapil, J. Sachin, IP Address Conflict Resolution and Configuration Theft Management, International Journal of Application or Innovation in Engineering & Management (IJAIEM), 3, 2, 2014.

[9] K.Ghassan, Android Wifi Network Management Tool By Using Simple Network Management Protocol, Diyala Journal of Engineering Sciences, 09, 04, 104-112, 2016.

[10] H. Soukaena, H. Hassan, Critical And Important Factors Related With Enhancing Wireless Communication Using Mimo Technology, Diyala Journal of Engineering Sciences, 08, 01, 42-63, 2015.

[11] H. Raad, Y. Hussien, K. Ghassan, Improvement Of Mimo System Using Sttc With Artificial Neural Network, Diyala Journal of Engineering Sciences, 10, 02, 01-11, 2017.