TASK QUARTERLY 10 No 3, 339–346

LAN INTERCONNECTION UNIT BASED ON AN ARTIFICIAL NEURAL NETWORK

HAMID ABDULLAH JALAB

Faculty of Science, Computer Science Department
Sana'a University,
P.O. Box 14526, Sana'a, Republic of Yemen
hamidjalab@yahoo.com

(Received 5 April 2006)

Abstract: This paper presents the design of an intelligent interconnection unit based on an artificial neural network (ANN), used when two local area networks (LAN) with different IEEE 802 standard protocols are connected. The proposed ANN is used to activate execution of suitable procedures bridging 802.X LAN and 802.Y LAN.

Keywords: computer networks, LAN bridge, neural networks, radial basis function (RBF)

1. Introduction

A single Local Area Network (LAN) proved to be insufficient to meet the requirements of an organization. It was thus deemed necessary to provide a device that would allow LAN's to be connected. Such device is called a bridge. In this paper, Artificial Neural Networks (ANN's) are used to design protocol conversion to connect two LAN's of different IEEE 802 standards.

A network system consists of hardware and software. Network software consists of protocols, or rules by which processes can communicate. Most networks support protocol hierarchies, with each layer providing services to the layers above it and insulating them from the details of protocols used in the lower layers.

Protocol stacks are typically based on the Open System International (OSI) model [1]. The OSI model was produced by the International Standard Organization (ISO). The logical structure of the OSI model is made up of seven protocol layers, as shown in Figure 1. The function of each layer is specified formally as a protocol that defines the set of rules used by the layer to communicate with a similar peer in another (remote) system. Each layer provides a defined set of services to the layer immediately above it. It also uses the services provided by the layer immediately below it to transport the message units associated with the protocol to the remote peer layer.

Devices that interconnect LAN's may operate at any one of several layers of the OSI model; those that operate in the data link layers (DLL) are called „bridges".

Application
Presentation
Session
Transport
Network
Data link
Physical

Figure 1. The OSI layers

A bridge is a design for use between local area networks (LAN's) that use identical protocols for the physical link layers (*e.g.* all conforming to IEEE 802.3). As a bridge is used in when all the LAN's share the same characteristics, the reader may wonder why one large LAN is not used instead? Depending on the circumstances, there are several reasons for the use of multiple LAN's connected by bridges:

- **Reliability:** The danger in connecting all data processing devices in an organization to one network is that a fault in the network may disable communication for all devices. When using bridges, the network can be partitioned into self-contained units.
- **Performance:** In general, performance of a LAN declines with increasing number of devices or length of wiring. A number of smaller LAN's will often offer improved performance if devices can be clustered so that the intranetworking traffic significantly improves.
- **Security:** The establishment of multiple LAN's may improve the security of communications. It is desirable to keep different types of traffic that have different security needs (*e.g.* accounting, personnel, strategic planning) on physically separate media. At the same time, different types of users of different security levels need to communicate through controlled and monitored mechanisms.
- **Geography:** Clearly, two separate LAN's are needed to support devices clustered in two geographically distant locations. Even in the case of two buildings separated by a highway, it may be far too easier to use a microwave bridge link than to attempt stringing a coaxial cable between the two buildings [2].

2. Networking of different LAN types

2.1. The interconnection problems

Many problems may be encountered when trying to use a bridge to connect two different IEEE 802 LAN's [2].

2.1.1. Frame format

Because of different transmission modes of the three basic LAN types, they all have different frame formats, as shown in Figure 2. There is no compelling technical reason for this incompatibility other than reluctance of the corporations supporting the three standards (*i.e.* Xerox, GM and IBM) to make an adjustment. As a result, any copying between different LAN's requires reformatting, which consumes CPU time, requires a new check sum calculation and introduces the possibility of undetected

<table>
<tr><td>802.3</td><td>PREAMBLE</td><td colspan="2">SD</td><td>DA</td><td>SA</td><td>LENGTH</td><td>DATA</td><td colspan="4">FCS</td></tr>
<tr><td>802.4</td><td>PREAMBLE</td><td>SD</td><td>FC</td><td colspan="2">DA</td><td>SA</td><td>DATA</td><td colspan="2">FCS</td><td colspan="2">ED</td></tr>
<tr><td>802.5</td><td>SD</td><td>AC</td><td>FC</td><td colspan="2">DA</td><td>SA</td><td>DATA</td><td>FCS</td><td colspan="2">ED</td><td>FS</td></tr>
</table>

←—— Header Frames ——→←———— Frame Contents ————→←—— Trailer ——→

SD = Start delimiter
FC = Frame control
AC = Access control
DA = Destination address
FCS = Frame check sequence
ED = End delimiter
FS = Frame status
SA = Source address

Figure 2. IEEE 802 LAN's frame formats

errors due to bad bits in the bridge memory. None of this would have been necessary if a single format has been agreed.

2.1.2. Bit rate

Interconnected LAN's do not necessarily run at the same data rate. A range of transmission bit rates is used with LAN's, including those shown in Table 1, [3].

If frames are received on a slow segment and are to be forwarded to a faster segment, there is no problem. If the reverse is the case, especially when the LAN is very busy, a problem can arise if frames have to be queued at the output port associated with the slower LAN. This is true even if the two LAN segments are of the same type. For example, if two token bus LAN segments are bridged, one operating at 10Mbps and the other at 5Mbps, acceleration of frames occurs during periods of heavy traffic. Since the amount of available memory is limited, the bridge starts to discard frames as insufficient storage memory is available. Although in practice the transport protocol entities in the affected source stations will initiate retransmission of another copy of these frames, the long timeout associated with this action means that the transit delay of frames will increase considerably. Moreover, there is no guarantee that the new copies will not experience a similar fate.

Table 1. LAN types' bit rates

LAN type	Available bit rate
CSMA/CD-Ethernet – 802.3	1, 2, 10Mbps
Token Bus – 802.4	1, 5, 10Mbps
Token Ring – 802.5	1, 4, 16Mbps

2.1.3. Frame size

The three LAN types each use a different maximum frame size: 802.3 uses 1518 bytes, 802.4 uses 8191 bytes, and that used with 802.5 is determined by the ring size (there is no upper limit, except that a station may not transmit longer than the token holding time with a default value of 10msec, the maximum frame length is 500 bytes). A serious problem may arise if a frame is first transmitted on, say, an 802.3 segment. Assuming that maximum frame sizes are used, the only way this can be overcome is for the bridge at the 802.3 segment to divide the frame into smaller subunits prior to transmission, each with the same destination and source address. Although this can be done, the so-called segmentation is not part of the 802.1 standard and so bridges do not normally offer this function.

2.2. Possible combination of interconnections

Because of the various formats for each frame size and bit rate, there are six possible combinations when bridging from IEEE 802.X to IEEE 802.Y, each with its own set of problems.

- **From 802.4 to 802.3:** There are two problems. First, 802.4 frames carry priority bits absent from 802.3 frames. As a result, if two 802.4 LAN's communicate via an 802.3 LAN, the priority will be lost by the intermediate LAN. The other problem is caused by a specific feature of 802.4: temporary token hand off. It is possible for 802.4 frames to have a header bit set to 1 to temporary pass the token to the destination, to let it send an acknowledgment frame. However, if such a frame is forwarded by a bridge, what should the bridge do? If it sends an acknowledgment frame itself, it will lie, as the frame has not been actually delivered yet. In fact, the destination may be dead. At the same time, if it generates no acknowledgement, the sender will almost certainly conclude that the destination is dead and report back failure to its superiors. The problem seems to have no solution.
- **From 802.5 to 802.3:** The 802.5 frame format has address-recognized bits (A) and frame-copied bits (C) in the frame status bytes. These bits are set by the destination to tell the sender whether the addressed station saw the frame and whether it copied it. Here again, the bridge can lie and report that the frame has been copied, but if it later turns out that the destination is down, a serious problem may arise. Again, a solution to this problem is hardly imaginable.
- **From 802.3 to 802.4:** The problem is what to put in propriety bits. A good case can be made for having the bridge retransmit all frames at the highest priority, as they have probably suffered enough delay already.
- **From 802.5 to 802.4:** The same problem with A and C bits as in the case "From 802.5 to 802.3" described above. Also, the definition of priority bits is different for the two LAN's, but beggars can't be choosers. At least the two LAN's have the same number of priority bits. All a bridge can do is copy the priority bits across and hope for the best.
- **From 802.3 to 802.5:** The bridge must generate priority bits, but there no other special problems.
- **From 802.4 to 802.5:** There is a potential problem with frames that are too long and the token hand off problem.

Table 2 summarizes the procedures which must be achieved to solve various problem occurring when bridging from IEEE 802.X to IEEE 802.Y LAN's.

2.3. The procedures

(1) Reformat the frame and compute the checksum.
(2) Reverse the bit order.
(3) Copy the priority, meaningful or not.
(4) Generate a fictitious priority.

Table 2. Problems when bridging 802.X with 802.Y

		Destination LAN		
Source LAN		802.3	802.4	802.5
	802.3	None	1, 4	1, 2, 4, 8
	802.4	1, 4, 8, 9, 10	None	1, 2, 3, 8, 9, 10
	802.5	1, 2, 5, 6, 7, 10	1, 2, 3, 6, 7	None

(5) Discard the priority.
(6) Drain the ring.
(7) Set A and C bits.
(8) Worry about congestion (fast LAN to slow LAN).
(9) Worry about token hand off acknowledgement (ACK) is either delayed or becomes nonexistent.
(10) Panic if frame is too long for the destination LAN.

2.4. Artificial Neural Network

In this paper, an Artificial Neural Network (ANN) is used for networking of different IEEE 802 LAN types. The function of the ANN is to activate the suitable procedures when using this unit to connect various LAN's of different IEEE 802 standards. The ANN has been trained using Radial Basis Function (RBF) networks. The advantages of using an RBF neural network are as follows:

(1) it trains faster than other types;
(2) when used for classification and decision problems, it leads to better decision boundaries than other types.

An RBF network is a feed-forward structure with a modified hidden layer and a training algorithm, which may be used for mapping. RBF networks emulate the behavior of certain biological networks [3]. Basically, a single hidden layer consists of locally tuned or locally sensitive units, while the output layer consists of linear units. In hidden-layer units, the unit response is localized and decreases as a function of distance of inputs from the unit's receptive field center. The overall network structure is shown in Figure 3.

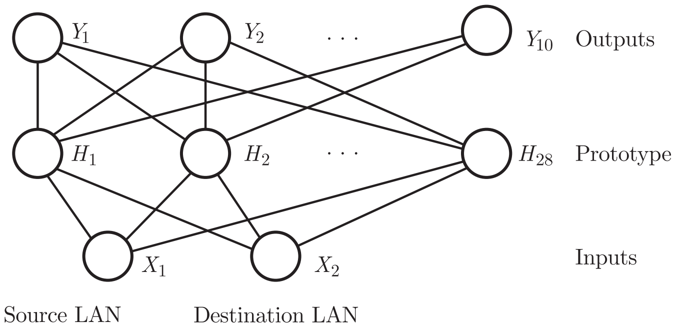

Figure 3. RBF network

2.5. Architecture

The RBFN used in this paper consists of three layers:

(1) input layer
Consists of 2 nodes representing the type of IEEE 802 standard. One node represents the IEEE 802 standard type of the source LAN, while the other node represents the IEEE802 standard type of the destination LAN;

(2) prototype layer
Consists of 28 nodes;

(3) output layer
Consists of 10 nodes representing the output nodes used to activate suitable procedures.

2.6. Training

The first phase of the network's training is a clustering phase, in which the weights incoming to the prototype layer become the centers of input vectors' clusters. The initial learning phase of a radial function-based network is an unsupervised clustering phase. The second learning phase of an RBFN is supervised learning, where the delta rule is used to determine weights from the prototype layer to output units [4, 5].

2.6.1. Self-Organizing Mapping (SOM) rule to determine RBF unit centers

(1) Initialize weights with small random values.

(2) Calculate distances from new input to all units as:

$$D_j = \sum_i (W_{ij} - X_i)^2,$$

where X_i is the input and W_{ij} is the units' weight.

(3) Select unit c as the winner, such that:

$$c = \min(D_j).$$

(4) Update the weights of the winning unit:

$$W_{ij}(t+1) = W_{ij}(t) + \alpha(X_i - W_{ij}(t)).$$

(5) Repeat steps 2 through 4 for the number of times equal to the number of input vectors in the cluster.

(6) Repeat step 5 several times (until all input vectors have been classified properly).

2.6.2. Learning rule for weight from cluster units to output units

The delta rule is used to determine weights from the prototype layer to output units.

(1) Apply an input vector, X_i, and its corresponding output vector, Y_k,to the X and Y inputs of the RBFN, respectively.

(2) Determine the wining prototype layer units.

(3) Update weights on the connections from the wining unit to the output units:

$$W_{jk}(t+1)=W_{jk}(t)+\beta(Y_k-W_{jk}),$$

where Y_k is the desired output (target) and W_{jk} is the weight on the connection from the j^{th} prototype layer.

(4) Repeat steps 1 through 3 until all vectors of all classes have mapped to satisfactory outputs.

3. Results and discussion

The proposed unit has been tested after fixing to acceptable values of α and β, in order to find the best numbers of training iterations, as well as the numbers of prototype neurons. The tests' results are shown in Figures 4 and 5.

The variation of output error is shown in Figure 4 as a function of the number of training iterations, when the number of prototype neurons equals 28, while α and β equal 0.1 and 0.5, respectively. The error decreases widely approaching 0 as the number of training iterations is increased.

In Figure 5, the variation of output error is shown as a function of the number of prototype layers, when the number of training iterations equals 20 000, while α and β equal 0.1 and 0.5, respectively. Here, the error decreases with increasing number of neurons in the prototype layer until 28 neurons; after a sharp increase occurs due to the ANN's oscillation. The number of neurons in the prototype layer should be fixed mainly in the function of the two main criteria, *viz.* the calculation time and accuracy [6].

Table 3 includes the best parameters used for the RBF neural networks in accordance with the above mentioned results.

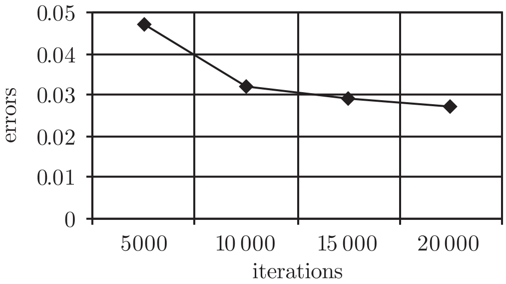

Figure 4. Error variation against training iterations

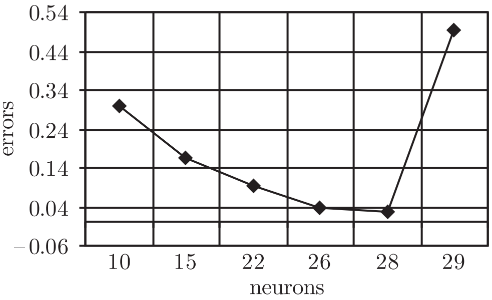

Figure 5. Error variation against number of neurons in the prototype layer

Table 3. RBF training parameters for the ANN

Activation function used	Sigmoid
Training iterations	20 000
Prototype neurons	28
Learning rate α	0.1
Learning rate β	0.5

The output test results of the neural network are shown in Table 4. The value of the ANN's output defines the procedure number which must be used according to the LAN type. For example, when 802.3 and 802.4 are connected (the second row of the table) the output shows logic 1 for both Y_{10} and Y_7. These outputs define the number of procedures to be used.

Table 4. The output simulation results

Y_{10}	Y_9	Y_8	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1
0.027072	0.027147	0.027102	0.027096	0.027111	0.027071	0.027088	0.027074	0.027135	0.027142
0.972860	0.027104	0.027148	0.972922	0.027145	0.027104	0.027128	0.027050	0.027133	0.027034
0.972919	0.972884	0.027117	0.972855	0.027128	0.027107	0.027145	0.972900	0.027102	0.027124
0.972850	0.027147	0.027138	0.972860	0.027130	0.027127	0.027080	0.972844	0.972843	0.972920
0.023314	0.023260	0.023261	0.023331	0.023312	0.023288	0.023288	0.023266	0.023291	0.023312
0.972916	0.972898	0.972885	0.027080	0.027158	0.027083	0.027069	0.972861	0.972925	0.972872
0.972910	0.972838	0.027151	0.027077	0.972897	0.972902	0.972854	0.027111	0.027121	0.972869
0.972887	0.972833	0.972923	0.027101	0.027166	0.972888	0.972844	0.027113	0.027109	0.027133
0.027087	0.027151	0.027073	0.027125	0.027109	0.027138	0.027065	0.027147	0.027099	0.027131

4. Conclusions

This LAN intelligent interconnection unit has been designed and implemented using Radial Basis Function (RBF) networks. The proposed unit has been tested for various numbers of training iterations and prototype neurons. Artificial Neural Networks have been used to activate execution of suitable procedures when LAN's with different protocols are connected. The simulation results have shown that the ANN gives exact results for solving connection problems of different LAN's.

References

[1] Stallings W 2002 *Data & Computer Communications*, 6th Edition, Prentice Hall of India
[2] Hamid A J and Mutea Al-Umari 2003 *Proc. of Yemen Sci. Conf.*, Seiyun, pp. 446–453
[3] Halsall F 1996 *Data Communications, Computers Networks and Open Systems*, 4th Edition, Addision-Wesley
[4] Schalkoff R J 1997 *Artificial Neural Networks*, McGraw-Hill International Editions
[5] Freeman J A and Skapura D M 2002 *Neural Networks Algorithms, Applications, and Programming Techniques*, 5th Edition, Pearson Education, Delhi
[6] Basulaim K O 2003 *Yemen Eng.* **3** (1) pp. 192–196