Applied Mathematical Sciences, Vol. 10, 2016, no. 19, 931 - 945
HIKARI Ltd, www.m-hikari.com
http://dx.doi.org/10.12988/ams.2016.510651

The Multi-Constrained Least-Cost Multicast Problem with Neural Networks in Fixed Time

N. Saber

Laboratory: Signals, Distributed Systems and Artificial Intelligence (SSDIA)
ENSET, Hassan II University, Casablanca
Bd Hassan II, Mohammedia, Morocco

M. Mestari

Laboratory: Signals, Distributed Systems and Artificial Intelligence (SSDIA)
ENSET, Hassan II University, Casablanca
Bd Hassan II, Mohammedia, Morocco

A. Ait El Mahjoub

ENSAM, Hassan II University,Casablanca
Avenue Mohamed Bouziane, Casablanca, Morocco

Abstract

In this paper, we propose a neural network architecture for the multicast routing algorithm, which find a multicast routing tree with minimal cost that satisfy constraints such as bandwidth and delay.The proposed network consists only of two kinds of neurons, and its processing time remains constant and independent of the input network size.

Keywords: Multi-constrained problem,neural networks, linear neuron, threshold-logic neuron, MAXNET, sorting network

1 Introduction

Multicast routing has become a major theme in today's information technology, and has attracted the attention of several researchers, because of the tremendous amount of activity in the area of multimedia communications, file sharing, interactive games, videoconferencing, on-demand video, radio and TV transmission ...etc

Multicast consists of transmitting simultaneously a message from a single source to multiple destinations, taking into account QoS (Quality of service) requirements, such as bandwidth, end-to-end delay, delay jitter, packet loss ratio, cost ..., etc. To guarantee an optimal diffusion, it is necessary to determine a minimal tree connecting the multicast nodes. In graph theory, this problem is known as the minimal Steiner tree problem, and has been shown to be NP-complete [22]. Several methods are proposed in the literature to solve routing problem. Chow [2] and Salama et al. [20] proposed two exact algorithms to solve this problem, but they are not viable in very large networks, because of their high degree of computational complexity. Heuristics proposed by Kompella et al. [10], Widyono [23], Parsa et al. [15] are the famous methods used to solve multicast routing problem , because they construct a feasible solution within reasonable time.

Neural networks are also proposed to solve the multicast routing problem, because of the important property of parallelism that they offer. Rauch and Winarske proposed a modification of the neural networks traveling salesman algorithm to solve routing problem [17]. In 1982, Hopfield presented the Hopfield neural network [5], since then, many researchers have been exploring HHNs. Pornavalai et al. [16] proposed a modification of HNNs to solve constrained multicast routing, but Gee and Prager [4] demonstrated that they are not efficient in large networks. Nozawa [14] and Jain and Sharma [6] proposed a modification of HNNs to solve multicast problem by adding other properties. Wang et al. [21] proposed the transiently chaotic neural networks to this problem.

In this paper, we propose a construction of multi-constrained least cost multicast tree based on neural networks, by using only two kinds of neurons: linear and threshold-logic neurons.These neurons have been used in constructing various kinds of neural networks : Cichoki and Unbehauen [3], Hopfield [5], Mestari et al. [12], Mestari [13], Khouil et al. [7], [8], [9], Saber et al. [18] [19]. Among all the neurons proposed in the literature, they are probably the easiest to implement in hardware. This paper is organized as follows: section 2 will give a formal definition of the problem of multicast routing, section 3 will present some basic functions, a description of our method to construct the multicast tree will be given in section 4, and in section 5 we will propose an implementation of this algorithm based on neural networks and section 6 will

present conclusion.

2 Mathematical notations

A network is modeled as a non-directed graph $G = (V, E)$, where V is a finite set of nodes, and E is the set of edges representing connections between network nodes. $e_{ij} = (v_i, v_j)$represent an edge from node v_i to node v_j, where $v_i \in V, v_j \in V$ and $v_i \neq v_j$.
For each edge e_{ij} we associate three non-negative functions: Cost: $C(e_{ij}) : E \to \mathbb{R}^+$ that represents the cost of the edge e_{ij}, delay: $D(e_{ij}) : E \to \mathbb{R}^+$ that represents the delay of transmission of one bit from node v_i to node v_j,and bandwidth: $B(e_{ij}) : E \to \mathbb{R}^+$ that represents available bandwidth on edge e_{ij}. We denote by $s \in V$ the source node,U the set of destination nodes and $d \in U$ a destination node. A multicast tree T is a sub-graph of G that spanning the multicast group $M = s \cup U$ and may contains Steiner nodes that not belong to the multicast group M. In the next definitions, we use e to designate a given node.

Definition 2.1. *The total cost of the tree T is defined as the sum of the cost of all edges in that tree and is given by:*

$$C(T) = \sum_{e \in T} C(e) \tag{1}$$

Definition 2.2. *The total delay of the path $P_T(s, d)$ is defined as the sum of the delay of all edges along $P_T(s, d)$ and is given by:*

$$D(P_T(s, d)) = \sum_{e \in P_T(s,d)} D(e) \tag{2}$$

Definition 2.3. *The bottleneck bandwidth of the tree T is defined as the minimum value of the bandwidth available over all the edges of the tree T and is given by:*

$$B(T) = \min\{B(e)/e \in T\} \tag{3}$$

Definition 2.4. *Let Δ_D be the delay constraint and B_{min} be the bandwidth constraint of the multicast tree.The constrained least-cost multicast problem is defined as :*

$$\begin{aligned} &minimize && C(T) \\ &subject \quad to && D(T) \leq \Delta_D \\ & && B(T) \geq B_{min} \end{aligned} \tag{4}$$

Definition 2.5. *To measure the quality of any multicast tree, we define the fitness vector* $f = (f_1, f_2)$ *where:*

$$\begin{cases} f_1 = \sum_{e \in T} C(e) \\ f_2 = \sum_{d \in M} max \ (0, \sum_{e \in P_T(s,d)} D(e) - \Delta_D) \end{cases} \tag{5}$$

f_1 measures the cost of the multicast tree and f_2 measures if the delay constraints are fulfilled. A multicast tree is good if f_1 is small and f_2 is zero [24].
In this manner, we can minimize the cost of the multicast tree and enforce fulfilling the delay constraints [24].

3 Some Basic functions

Let n_i be the number of edges adjacent to a given node v_i, where $i \in \{1, 2, ..., N\}$

Definition 3.1. *Let* e_{ij} *and* e_{iq} *be two edges adjacent to node* v_i*, with respective costs* $C(e_{ij})$ *and* $C(e_{iq})$ *. The comparison function of* $C(e_{ij})$ *and* $C(e_{iq})$ *is defined as follows:*
for $j < q$*:*

$$comp(C(e_{ij}), C(e_{iq})) \stackrel{def}{=} \begin{cases} 1 & if \quad C(e_{iq}) \geq C(e_{ij}) \\ 0 & if \quad C(e_{iq}) < C(e_{ij}) \end{cases} \tag{6}$$

and for $j > q$*:*

$$comp(C(e_{ij}), C(e_{iq})) \stackrel{def}{=} \begin{cases} 1 & if \quad C(e_{iq}) > C(e_{ij}) \\ 0 & if \quad C(e_{iq}) \leq C(e_{ij}) \end{cases} \tag{7}$$

Definition 3.2. *The order of the cost function* $C(e_{iq})$ *associated to a given edge* e_{iq} *in the list of edges adjacent to node* v_i *may be given by the following order function:*

$$ord(C(e_{iq})) \stackrel{def}{=} \sum_{j<q} comp(C(e_{ij}), C(e_{iq})) + \sum_{j>q} comp(C(e_{ij}), C(e_{iq})) + 1 \tag{8}$$

Definition 3.3. *Let* $k(1 \leq k \leq n_i)$ *a given number. The equality function determines whether the order of the cost* $C(e_{ij})$ *of edge* e_{ij} *is equal or not to* k*, and is defined as follows:*

$$eq[ord(C(e_{ij}))] \stackrel{def}{=} \begin{cases} 1 & if \quad ord(C(e_{ij})) = k \\ 0 & \quad otherwise \end{cases} \tag{9}$$

Proposition 3.4. *Let $C_{(k)}$ be the k^{th} largest cost function from costs associated to edges adjacent to node v_i, then :*

$$C_{(k)} = \sum_{j=1}^{n_i} C(e_{ij}).eq[ord(C(e_{ij})), k] \tag{10}$$

4 Description of the proposed method

In this section, we describe all the steps of the algorithm for multicast tree, which is based on the mathematical model. This algorithm mimics Kruskal's MST algoritm [11] and has 3 phases: initialize, generate and evaluate. Each phase will be described separately.

Phase 1 : initialize

inputs: $G = (V, E)$: network topology, $B(e)$: available bandwidth on edge e, B_{min} : QoS bandwidth constraint parameter.

output:

A graph $G' = (V', E')$

procedure:

$G' := \emptyset$

Remove from G all edges that violate minimum bandwidth constraint:

for e in G:

if $B(e) \geq B_{min}$ then $G' := G' \cup \{e\}$

Phase 2: generate

inputs : $G' = (V', E')$:the new network topology, $C(e_{ij})$: cost of transmission of edge e_{ij}, s: the source node, U : the set of destination nodes.

output:

A multicast tree T

procedure:

1) for each node $v_i \in V'$, dressing a list L_i that contains all nodes e_{ij} adjacent to v_i , where $j \in \{1, ..., n_i\}$;

2) for each list L_i , where $v_i \in V'$,sorting the elements e_{ij}, $j \in \{1, ..., n_i\}$ in ascending order of their costs;

3) initialize the multicast tree with the source node $T \leftarrow \{s\}$

4) remove s from all lists L_i ;

5) where U is not entirely included in T

   a. dressing a list L_T that contains all edges adjacent to T

   b. sorting the elements of L_T in ascending order of their costs;

c. if L_T contains an element $d \in U$, then
$T \leftarrow T \cup \{d\}$;
remove d from all lists L_i;
else $T \leftarrow T \cup \{t\}$ where t is the element with the minimum cost in L_T;
remove t from all lists L_i;

In the end of this phase, we obtain a single solution.To obtain a set containing a large number of solutions, we can iterate this phase to generate successive approximations of the multicast tree.
Phase 3: evaluate

1) For each multicast tree T, generated in the previous phase, calculate the fitness vector $f = (f_1, f_2)$ defined in equation (5)

2) Given two multicast trees T_1 and T_2 with respective fitness vectors $f' = (f_1^{'}, f_2^{'})$ and $f'' = (f_1^{''}, f_2^{''})$, T_1 is better than T_2 if and only if :

   a) $f_2^{'} < f_2^{''}$;or

   b) $f_2^{'} = f_2^{''}$ and $f_1^{'} < f_1^{''}$

5 Neural networks architecture

In this section, our mean concern is to implement our method directly in specialized hardware, by using only two kinds of neurons, that we will define in the next section.

5.1 neurons used

All neural networks proposed in this paper use only two kinds of neurons: the linear and the threshold-logic neuron.The only difference between these neurons is in their activation functions.
Let be n inputs $(x_1, x_2, ...x_n)$; Φ is an activation function; $\theta(\theta \in \mathbb{R})$ is the external threshold , also called an offset or biais; ω_i are the synaptic weights or strengths ; and y represents the output.

Definition 5.1. *The linear neuron sum the n weighted inputs and pass the result through a non-linearity according to*

$$y = \Phi_L(\sum_{i=1}^{n} \omega_i x_i - \theta) \tag{11}$$

Where Φ_L is the linear activation function, defined as:

$$\Phi_L(x) = x \tag{12}$$

with $x = \sum_{i=a}^{n} \omega_i x_i - \theta$

Definition 5.2. *The Threshold-logic neuron sum the n weighted inputs and pass the result through a non-linearity according to*

$$y = \Phi_T(\sum_{i=1}^{n} \omega_i x_i - \theta) \tag{13}$$

Where Φ_L is the binary activation function, defined as:

$$\Phi_T(x) = \begin{cases} 1 & if \quad x \leq O \\ 0 & otherwise \end{cases} \tag{14}$$

with $x = \sum_{i=a}^{n} \omega_i x_i - \theta$

5.2 Bandwidth constraint network

The function of the bandwidth constraint network is to compare available bandwidth $B(e)$ on edge e with the minimum bandwidth required by the service to run: if $B(e) < B_{min}$, the edge e can not be included in the multicast tree T. The bandwidth constraint network BCN is shown by the diagram in fig. 1

5.3 Order network

The function of the order network is to give the order of the cost $C(e_{iq})$ of any edge e_{iq} adjacent to node v_i. This network compute the order function given by equation (8)
The order function (8) is equivalent to:

$$ord(C(e_{iq})) = \sum_{j<q} \Phi_T(C(e_{iq}) - C(e_{ij})) + \sum_{j>q} \Phi_T(-\Phi_T(C(e_{ij}) - C(e_{iq}))) + 1 \tag{15}$$

Where Φ_T is defined by equation (14).
The order network is shown in fig.2, and it is denoted by ON_q (see [1] [13]).

5.4 Equality network

The equality network determines whether the order of an edge $C(e_{ij)}$ is equal or not to a given number, k.
This network compute the equality function given by equation (9).

for all $C(e_{ij})$ and $(1 \leq k \leq n_i)$,

$$eq[ord(C(e_{ij}))] \stackrel{def}{=} \begin{cases} 1 & if \quad ord(C(e_{ij})) = k \\ 0 & \quad otherwise \end{cases} \tag{16}$$

Proposition 5.3. *For all* $C(e_{ij})$ *and* $(1 \leq k \leq n_i)$:
$eq[ord(C(e_{ij})), k] = \Phi_T(-\Phi_T(ord(C(e_{ij})) - k - 1) - \Phi_T(k - ord(C(e_{ij})) - 1))$
where Φ_T *is defined in equation* (14)

The equality network consists of three threshold-logic neurons and is denoted by EN. Its schematic representation is shown by diagram in fig.3.(see [1] [13]).

5.5 Selection network

The function of selection network is to select from among the costs associated to edges adjacent to node v_i the cost function corresponding to the order fixed by the adjustment input and to transfer it to output.This network is composed of n_i equality networks, EN, followed by a linear neuron. the selection network is shown in fig.4 and is denoted as SN.It detects and transfers only the appropriate cost whose order equals k (i.e the k^{th} largest cost : $C_{(k)}$) to output (see [1] [13]).

5.6 MAXNET

MAXNET is shown in fig.5, where the decision maker determines which order is to appear at the output. For example, to output the maximum cost from a list of n_i elements, the order is n_i, or 1 to output the minimum value. If the input size is n_i, the number of neurons in MAXNET is $\frac{3}{2}n_i^2 + \frac{5}{2}n_i + 1$. The total processing time of MAXNET remains constant regardless the size of input list; this network consists of two kinds of neurons arranged in six layer, so the processing time is six times the processing time of a single neuron, and any modification in size of input list causes the modification in the number of neurons, not the layers themselves (see [1] [13]).

5.7 Sorting network

Sorting network is shown in fig.6, and consists of n_i MAXNET set up in parallel. The function of SORTNET is to giving the order of any element of the input list, and arranging them in ascending order.Sorting time remains constant irrespective to the the size of input list, and is equal to the processing time of a single MAXNET (see [1] [13]).

5.8 Fitness networks

The fitness networks FCN and FDN are shown respectively in fig.7 and fig.9. The network shown in fig.8 calculates the difference between the total delay of path $P_T(s,d)$ and the delay constraint. The function of these networks is to calculate the fitness vector $f=(f_1,f_2)$ of any multicast tree. Let T_1 and T_2 be two multicast trees, with respective fitness vectors $(f_1^{'},f_2^{'})$ and $(f_1^{''},f_2^{''})$.The network shown by fig. 10 compares T_1 and T_2:

$$comp(T_1,T_2) \stackrel{def}{=} \begin{cases} 1 & if \quad T_1 \; is \; better \; than \; T_2 \\ 0 & if \quad T_2 \; is \; better \; than \; T_1 \end{cases} \tag{17}$$

6 Conclusion

The multimedia multicast routing problem arises in many multimedia communication applications, and this problem has been proved to be NP-complete. In this paper, we proposed a new method for solving this problem in real time.This method was be implemented by neural networks, by using only two kinds of neurons: linear and threshold logic neurons.Among all the neurons proposed in the literature, they are probably the easiest to implement in hardware.All neural networks proposed in this paper have a simple configuration,and their processing time remains constant and independent of the input network size.Our approach used in the field of neural networks;our primary concern was how to organize neurons into a network so that it can solve a specific problem, with an emphasis on fully utilizing the massive parallelism property offered by neural networks.

References

[1] A. Badi, K. Akodadi, M. Mestari and A. Namir, Θ (1) Time Neural Network Minimum Distance Classifier and its Application to Optical Character Recognition Problem, *Appl. Math. Sci.*, **2** (2008), no. 26, 1253-1282.

[2] C.H. Chow, On multicast path finding algorithms, *IEEE INFCOM '91. The Conference on Computer Communications. Tenth Annual Joint Conference of the IEEE Computer and Communications Societies Proceedings,* **3** (1991), 1274-1283. http://dx.doi.org/10.1109/infcom.1991.147651

[3] A. Cichoki and R. Unbehauen, Neural Networks for solving systems of linear equations and related problems, *IEEE Trans. Circuits Syst. I: Fundamental Theory and Applications*, **39** (1992), 124-138. http://dx.doi.org/10.1109/81.167018

[4] A.H. Gee and R.W. Prager, Limitations of Neural Networks for Solving Traveling Salesman Problems, *IEEE Trans. Neural Networks*, **6** (1995), 280-282. http://dx.doi.org/10.1109/72.363424

[5] J.J. Hopfield, Neural Networks and Physical Systems with Emergent Collective Computational Abilities, *Proc. Nat. Academy of Sciences*, **79** (1982), 2554-2558. http://dx.doi.org/10.1073/pnas.79.8.2554

[6] S. Jain, J.D. Sharma, Delay bound multicast routing using hopfield neural network, *International Journal of Computer Theory and Engineering*, **2** (2010), 1793-8201. http://dx.doi.org/10.7763/ijcte.2010.v2.172

[7] M. Khouil, N. Saber, M. Mestari, Neural Network in Fixed Time for Collision Detection between Two Convex Polyhedra, *International Science Index*, **8** (2014), no. 5.

[8] M. Khouil, N. Saber, M. Mestari, Reseaux de neurones pour la detection de collision et localisation de contacts des polyedres convexes, *Revue Mediterraneenne des Telecommunications*, **4** (2014), no. 2.

[9] M. Khouil, N. Saber, M. Mestari, Collision Detection for Three Dimension Objects in a Fixed Time, *Third IEEE International Colloquium in Information Science and Technology (CIST)*, (2014), 235-240. http://dx.doi.org/10.1109/cist.2014.7016625

[10] V.P. Kompella, J.C. Pasquale, G.C. Polyzos, Multicast routing for multimedia communication, *IEEE/ACM Transactions on Networking*, **1** (1993), 286-292. http://dx.doi.org/10.1109/90.234851

[11] J.B. Kruskal, On the shortest spanning subtree of a graph and the traveling salesman problem, *Proceedings of the American Mathematical Society*, **7** (1956), 48-50. http://dx.doi.org/10.1090/s0002-9939-1956-0078686-7

[12] M. Mestari, M. Benzirar, N. Saber and M. Khouil, Solving Nonlinear Equality Constrained Multiobjective Optimization Problems Using Neural Networks, *IEEE Transactions on Neural Networks and Learning Systems*, **26** (2015), no. 10, 2500-2520. http://dx.doi.org/10.1109/tnnls.2015.2388511

[13] M. Mestari, An Analog Neural Network Implementation in Fixed Time of Adjustable Order Statistic Filters and Applications, *IEEE Transactions on Neural Networks*, **15** (2004), no. 3, 766-785. http://dx.doi.org/10.1109/tnn.2003.820656

[14] H. Nozawa, A Neural-Network Model as a Globally Coupled Map and Applications Based on Chaos, *Chaos: An Interdisciplinary Journal of Nonlinear Science*, **2** (1992), 377-386. http://dx.doi.org/10.1063/1.165880

[15] M. Parsa, Q. Zhu, J.J. Garcia-Luna-Aceves, An iterative algorithm for delay-constrained minimum-cost multicasting, *IEEE/ACM Transactions on Networking*, **6** (1998), 461-474. http://dx.doi.org/10.1109/90.720901

[16] C. Pornavalai, G. Chakraborty and N. Shiratori, A Neural Network Approach to Multicast Routing in Real-Time Communication Networks, *Proc. Int. Conf. Network Protocols*, (1995), 332-339. http://dx.doi.org/10.1109/icnp.1995.524849

[17] H.E. Rauch, T. Winarske, Neural Networks for Routing Communication Traffic, *IEEE Cont. Syst. Mag.*, **8** (1988), 26-31. http://dx.doi.org/10.1109/37.1870

[18] N. Saber, M. Khouil, M. Mestari, Delay Constrained Multicast Tree with Neural Networks in Fixed Time, *International Journal of Engineering Sciences and Research Technology,* (2014), 153-158.

[19] N. Saber, M. Khouil, M. Mestari, Neural Networks Based on Adjustable-Order Statistic Filters for Multimedia Multicast Routing, *Third IEEE International Colloquium in Information Science and Technology (CIST)*, (2014), 435-439. http://dx.doi.org/10.1109/cist.2014.7016660

[20] H.F. Salama, D.S. Reeves, Y. Viniotis, Evaluation of multicast routing algorithms for real-time communication on high-speed networks, *IEEE Journal on Selected Areas in Communications*, **15** (1997), 332-345. http://dx.doi.org/10.1109/49.564132

[21] L. Wang, S. Li, F. Tian and X. Fu, A noisy chaotic neural network for solving combinatorial optimization problems: Stochastic chaotic simulated annealing, *IEEE Trans. Syst., Man and Cybernetics, Part B (Cybernetics)*, **34** (2004), 2119-2125. http://dx.doi.org/10.1109/tsmcb.2004.829778

[22] Z. Wang, J. Crowcroft, Quality-of-service routing for supporting multimedia applications, *IEEE Journal on Selected Areas in Communications,* **14** (1996), 1228-1234. http://dx.doi.org/10.1109/49.536364

[23] R. Widyono, *The Design and Evaluation of Routing Algorithms for Real-Time Channels*, Technical Report TR-94-024, University of California, Berkeley, 1994.

[24] Q. Zhang and Y. W. Leung, An orthogonal genetic algorithm for multimedia multicast routing, *IEEE Trans. Evol. Comput.*, **3** (1999), no. 1, 53-62. http://dx.doi.org/10.1109/4235.752920

Received: October 30, 2015; Published: March 17, 2016

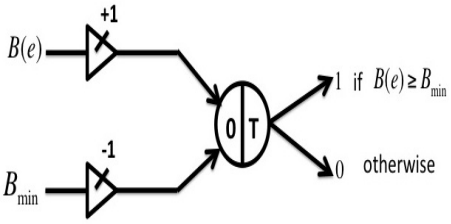

Figure 1: Bandwidth constraint network BCN

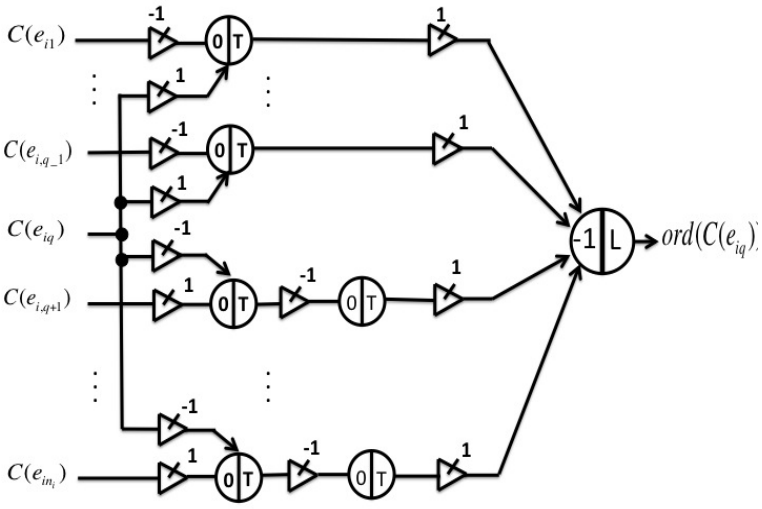

Figure 2: order network

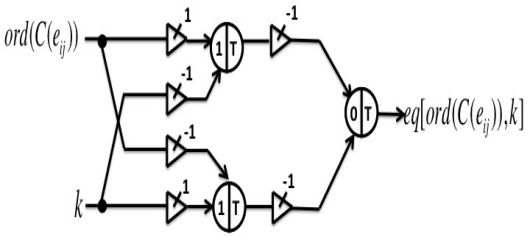

Figure 3: equality network

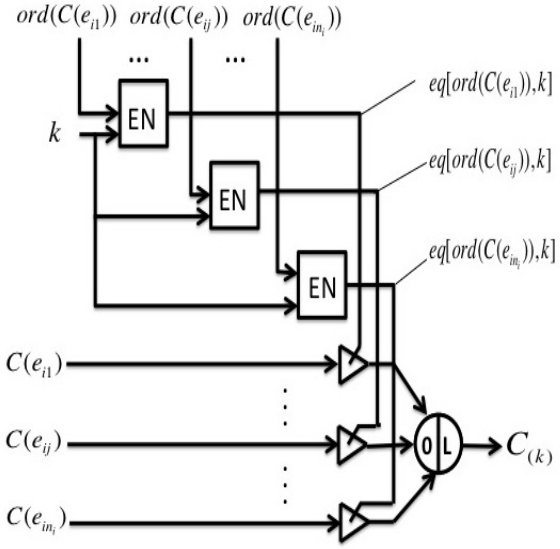

Figure 4: selection network

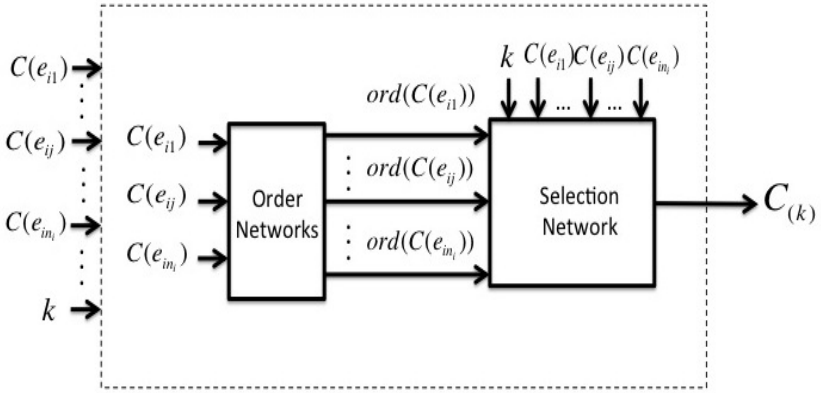

Figure 5: MAXNET

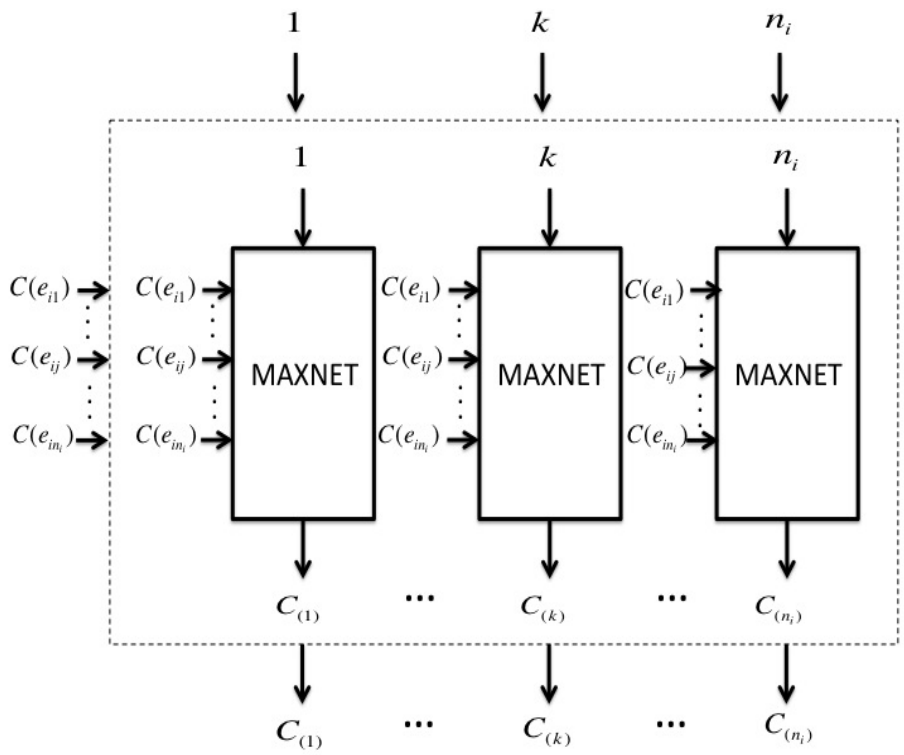

Figure 6: Sorting network

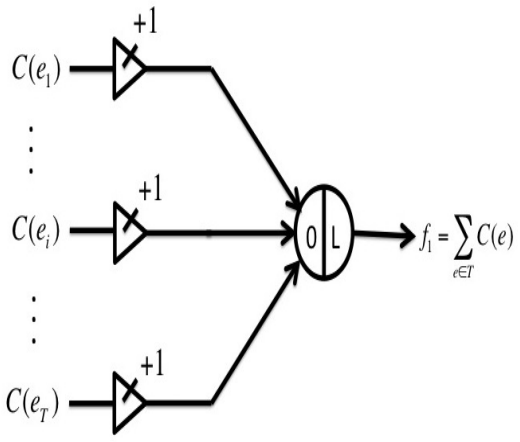

Figure 7: FCN

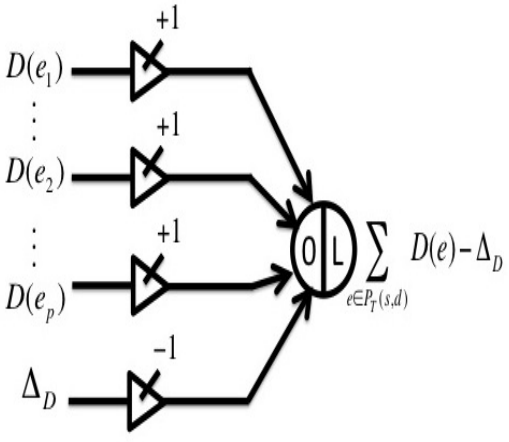

Figure 8:

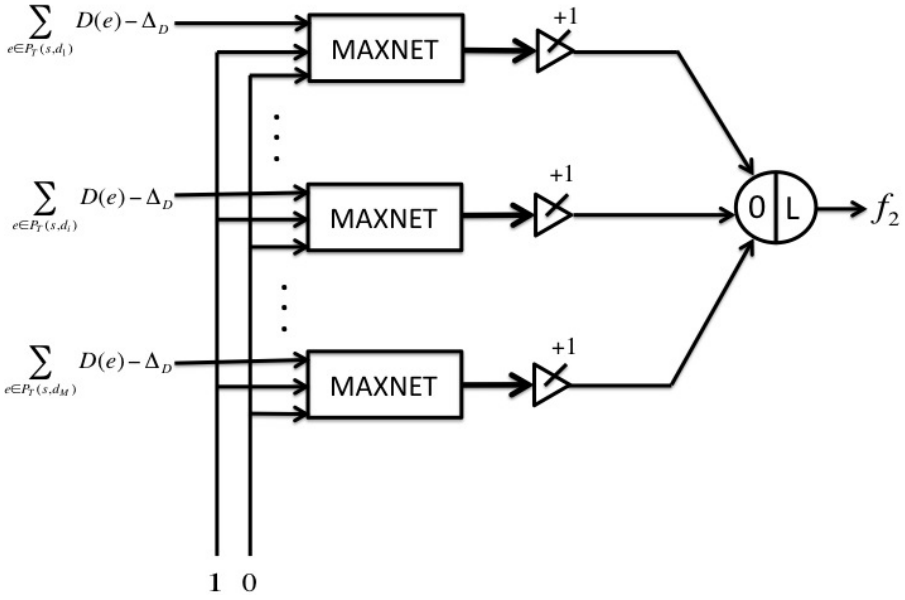

Figure 9: FDN

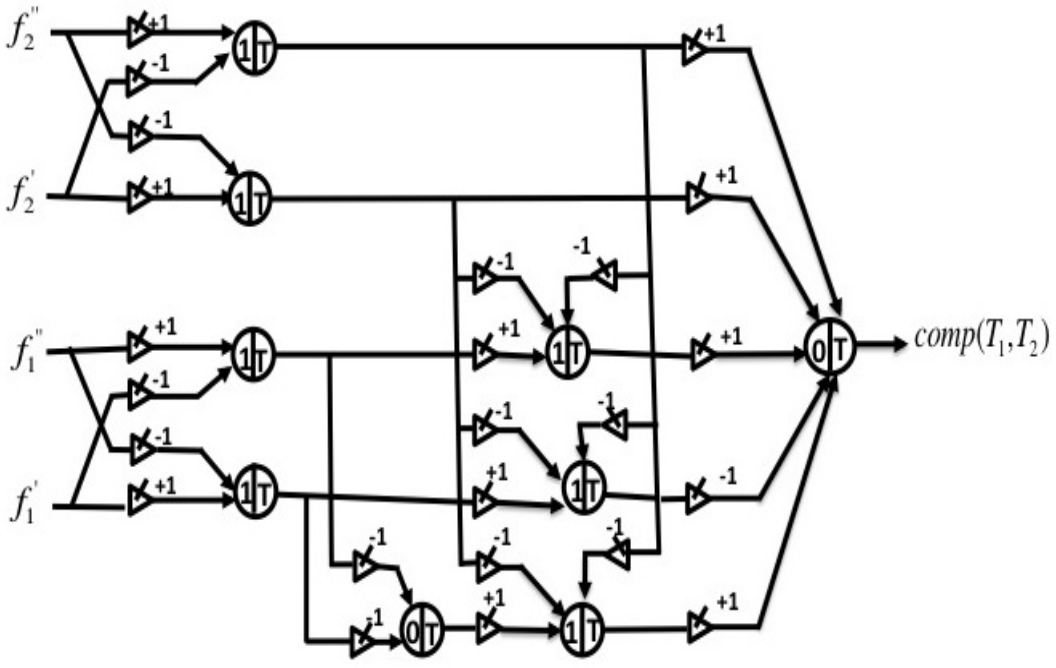

Figure 10: comparison network for two multicast trees