A CELLULAR NEURAL NETWORKS BASED DIFFSERV SWITCH FOR SATELLITE COMMUNICATION SYSTEMS

Romano Fantacci(1), Roberto Gubellini(1), Tommaso Pecorella(2), and Daniele Tarchi(1)

(1)Dipartimento di Elettronica e Telecomunicazioni
Università di Firenze
Via di Santa Marta, 3
50139 Firenze - Italy
email: {fantacci,gubellini,tarchi}@lenst.det.unifi.it

(2)CNIT - Unità di Ricerca di Firenze
Via di Santa Marta, 3
50139 Firenze - Italy
email: pecos@lenst.det.unifi.it

ABSTRACT

Recent developments of Internet services and advanced compression methods has revived interest on IP based multimedia satellite communication systems. However a main problem arising here is to guarantee specific Quality of Service (QoS) constraints in order to have good performance for each traffic class.

Among various QoS approach used in Internet, recently the DiffServ technique has became the most promising solution, mainly for its simplicity with respect to different alternatives. Moreover, in satellite communication systems, DiffServ policy computational capabilities are placed at the edge points (end-to-end philosophy); this is very important for a network constituted by one satellite link because it allows to reduce the implementation complexity of the satellite on-board equipments.

The satellite switch under consideration makes use of the Multiple Input Queuing approach. Packets arrived at a switch input are stored in a shared buffer but they are logically ordered in individual queues, one for each possible output link. According to the DiffServ policy, within a same logical queue, packets are reordered in individual sub-queues according to the priority. A suitable implementation of the DiffServ policy based on a Cellular Neural Network (CNN) is proposed in the paper in order to achieve QoS requirements.

The CNNs are a set of linear and nonlinear circuits connected among them that allow parallel and asynchronous computation. CNNs are a class of neural networks similar to Hopfield Neural Networks (HNN), but more flexible and suitable for solving the output contention problem, inherent of switching systems, for VLSI implementation.

In this paper a CNN has been designed in order to maximize a cost functional, related to the on-board switch throughput and QoS constraints. The initial state for each neural cell is obtained looking at the presence of at least one packet from a certain input logical queue to a specific output line. The input value for each neural cell is a function of priority and length of each input logical queue. The versatility of neural network make feasible to take the best decision for the packet to be delivered to each output satellite beam, in order to meet specific QoS constraints. Numerical results for CNN approach highlights that Neural network convergence within a time slot is guaranteed, and an optimal, or at least near-optimal, solution in terms of cost function is achieved.

The proposed system is based on the IETF (Internet Engineering Task Force) recommendations; this means that traffic entering the switching fabric could be marked as Expedited Forward (EF) or Assured Forward (AF), otherwise handled as Best Effort (BE). Two Assured Forward classes with different emission priority have been implemented, taking into account time spent inside the logical queue and its length. Expedited Forward traffic is typical of services to be delivered with the maximum priority, as streaming or interactive services. The packets, belonging to services that need a certain level of priority with low packet loss, are marked as Assured Forward. Best Effort traffic is related to e-mail or file transfer, or other that have not particular QoS requirements.

Proc. of First International Conf. on Advanced Satellite Mobile Systems (ASMS - 2003) 10-11 July 2003, Frascati, Italy (ESA SP-541, July 2003)

The CNN used to solve conflict situations act as an arbiter for all the output links. Differently from other Multiple Input Queuing approach, where one arbiter for each output line is present, in proposed approach there exist only one arbiter that make the best decision. The selected rule has been defined in order to give priority to packets, according to opportunely defined functionals characteristic of each traffic class, under the constraint that no more than one packet can be delivered to the same output line. The functionals depend on queue length and time spent inside the queue by front packet.

The performance of the proposed DiffServ switch has been derived in terms of delay and jitter; buffer occupancy has been analyzed for different configuration, such as a unique common buffer, one buffer for each input line, one buffer for each input line and each priority class.

The obtained results highlight an high flexibility of satellite switch with CNN, taking into account that functional used to calculate priority of each queue could be easily changed, without any complexity gain nor change in CNN structure, in order to consider different traffic characteristic. Numerical results show that proposed algorithm outperform the switches based on Multiple Input Queuing, that use strictly priority methods, in terms of delay and jitter. Different buffer size have been also considered in order to analyze packet loss for CNN switch algorithm, comparing different configuration described above.

The good behavior of the proposed DiffServ switch has been verified in the case of traffic with pareto distribution for packet length and a geometrical distribution for packet interarrival time, highlighting good performance in terms of delay and jitter. Numerical results also demonstrate the stability of this method for heavy load traffic; in particular maximum permitted load is higher for higher priority classes.

1. INTRODUCTION

Due to wide coverage and asymmetric bandwidth, satellite communications are becoming an important part of global multimedia networks. Actual trend in global communications is to integrate satellite systems in the development of Internet [1]. A lot of services are being created exploiting wide diffusion of Internet, however results are not always acceptable due to different requirements of each service. In particular each network-based application has different requirements in terms of delay, jitter, bandwidth and packet loss. Nowadays, Internet is a Best Effort network and each packet is processed as quickly as possible, without any guarantee on delivery time. The delay introduced by each network node has also a high variability. The expansion of Internet to a commercial infrastructure requires the development of new protocols, architectures and network systems in order to guarantee different Quality of Service (QoS) levels to various applications according to their needs. The introduction of QoS-driven policy in a satellite system requires the development of advanced solutions in terms of complexity and cost due to reduced on-board capabilities.

The Internet Engineering Task Force (IETF) has proposed many service models and mechanisms in order to meet the demand for QoS; in this paper we consider the Differentiated Services (DiffServ) approach [2] that is scalable and keep low complexity core nodes, like satellites, moving complexity to the edge nodes. This approach uses the DS Code Point field of the IPvX packet header in order to identify the payload type. Each node manages the packets according to this field; in DiffServ systems this operation is known as PHB (Per Hop Behavior). The satellite system considered in this paper is geostationary (GEO) and multibeam. For our purpose, the orbit type modify the average delay and jitter without any change to the structure of the switch. The presence of more than one input beam (uplink) and one output beam (downlink) requires procedures to avoid packet collision; at the same time we have implemented a mechanism to supply QoS. Previous techniques are able to have a complete control over switching matrices hardware-based without introducing QoS requirements in solving the switching problem [3]. Our approach is to use neural networks, for their flexibility and easy adaptive structure that let introduce QoS constrains without any change to the basic structure of the neural controller.

2. THE SWITCHING PROBLEM

Packet switching operation from the input lines to the output lines require to choose which packets have to be transmitted, reading the IP destination and the DS Code Point field. Introducing this operation in actual satellites, that have on-board processing capabilities, requires to discontinue typical switching fabric.

An input buffer approach with a FIFO queue for each input suffers the so called Head of Line (HOL) blocking problem [4]; the first packet of a queue could block the other packets of the same queue also if the destination port of these one is free. Solving this problem using an Output buffering approach require to have a switching speed that grow with the dimension (N) of the system and that can manage up to N packets in a time slot [5]. Multiple Input Queuing with Input Shared Buffer architecture like the one in Fig. 1 seems to be the best solution allowing to reach high throughput

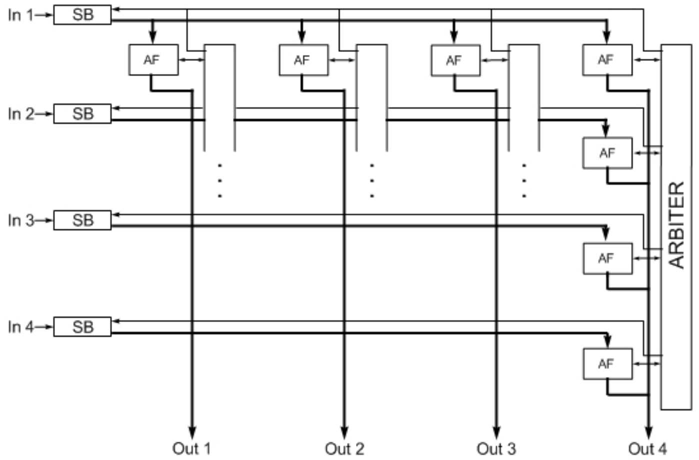


Figure 1: Multiple Input Queuing with Input Shared Buffer.

maintaining switching speed equal to the input line packet-rate [3]. In particular, Multiple Input Queuing solution foresee the use of an arbiter for each output line to choose which packet has to be sent. Each arbiter is independent and do not communicate with the others; this solution allows to extract until N packets from the same input line, requesting reading speed of input buffers N times faster than reading speed of output buffers, making this solution not feasible for a satellite implementation. On the other hand, arbiter decision is based on a strictly priority queue scheme, and the low priority packets are blocked until there are packets of higher priority, resulting in the starvation problem. Introducing flexible arbiter in this structure is very complex; we propose the use of a neural network and particularly a Cellular Neural Network in order to solve the problems of the Multiple Input Queuing.

3. CELLULAR NEURAL NETWORK

The Cellular Neural Networks (CNN) [6, 7] are an evolution of the Hopfield Neural Networks (HNN) [8] capable of parallel analogical computing in real time and belonging to the class of single layer neural networks with feedback. This type of neural networks overcome some limitations of HNN and they are also well suited for VLSI implementation. The basic unit of a CNN is called a cell and it is made by linear and non linear circuits like capacitors, resistors, controlled and independent sources (Fig. 2). Each cell is connected only to its neighbor cells according to a regular structure; each one affects all the others due to the propagation effects of the continuous time dynamics of CNN. Let consider an $M \times N$ cellular neural network having M rows and N columns (Fig. 3). The circuit equations of a cell, derived applying KCL (Kirchhoff's Current Law) and KVL (Kirchhoff's Voltage Law) are:

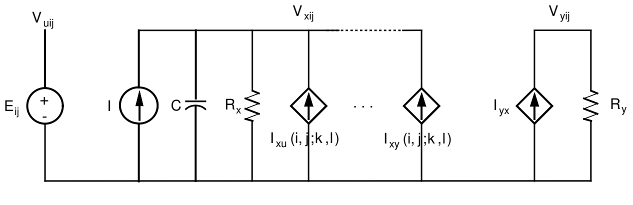


Figure 2: An example of a cell circuit.

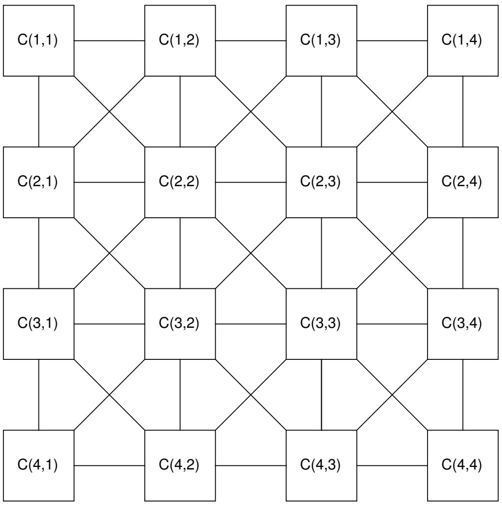


Figure 3: A two dimensional Cellular Neural Network (M=N=4).

- State equation (voltage v_{xij}):

$$C\frac{dv_{xij}(t)}{dt} = -\frac{1}{R_x}v_{xij}(t) + \sum_{C(k,l)\in N_r(i,j)} A(i,j;k,l)v_{ykl} + \sum_{C(k,l)\in N_r(i,j)} B(i,j;k,l)v_{ukl} + I, \quad 1 \le i \le M; \quad 1 \le j \le N; \tag{1}$$

- Input equation (voltage v_{uij}):

$$v_{uij} = E_{ij}, \qquad 1 \le i \le M, \quad 1 \le j \le N; \tag{2}$$

- Output equation (voltage v_{yij}):

$$v_{yij}(t) = \frac{1}{2}(|v_{xij}(t)+1| - |v_{xij}(t)-1|), \qquad 1 \le i \le M; \quad 1 \le j \le N. \tag{3}$$

In (1) the terms related to the input and output voltage of the cell belong to the neighbor of the cell (i,j) ($C(i,j)$) that can be, for example, a circle or a row or a column. On the other hand the physical dynamic range of a CNN can be computed by the following formula:

$$v_{\max} = 1 + R_x\,|I| + R_x \max_{\substack{1\le i\le M\\ 1\le j\le N}} \left[\sum_{C(k,l)\in N_r(i,j)} (|A(i,j;k,l)| + |B(i,j;k,l)|) \right] \tag{4}$$

In order to choose suitable parameters of (1), let us estimate the upper bound of the dynamic range for the implementation of the circuit.

4. CNN BASED SWITCHING SYSTEM

Let assume a system with N input and output lines and capable of managing P different classes of traffic. The proposed CNN is rectangular with M rows, where $M = NP$, and N columns; each cell is an arbiter for one queue located at each cross point between an input line and an output line of a specific priority (Fig. 4). Each queue is logically divided in several sub-queue related to different traffic classes; in particular in this paper we have assumed 4 traffic classes (or PHB

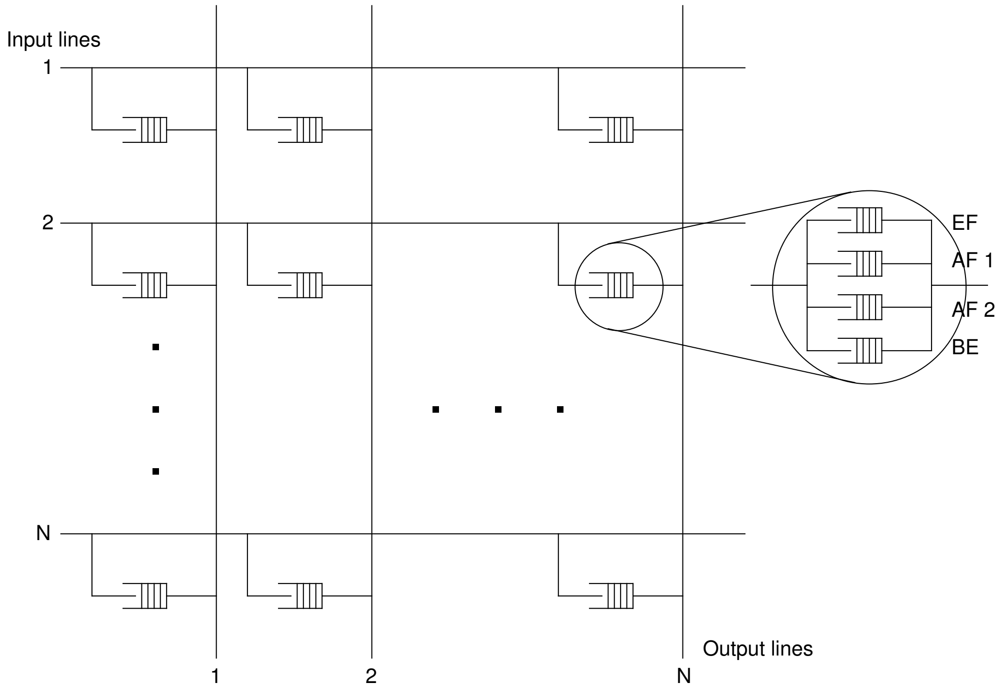


Figure 4: Logical structure for CNN switch.

using a DiffServ terminology) and precisely one Expedited Forward class [9], two type of Assured Forward classes [10] with different priority and one Best Effort class.

In the considered problem the neighborhood of a cell are all the cells belonging to the same row and column. The neighborhood has this shape because of the constraint to respect. In particular it contains the cells of the same column in order to avoid the collision of packets in output. Moreover, the neighborhood has also been extended to cells belonging to the same row in order to restrict the maximum number of packets routed from the same input line and the same traffic class; otherwise Multiple Input Queue with Shared Input Buffer requires input lines capable of routing a number of packets equal to the number of output lines. This requirement could be hard to respect for large systems. The introduction of this constraint and the presence of a unique arbiter (the CNN itself) for the whole system overcome the lack of cooperation between the N arbiters of the classical Multiple Input Queuing solution.

Thus, supposing a normalized value for capacitors ($C = 1F$) and resistors ($R_x = 1\Omega$), the state equation of the cells of CNN could becomes:

$$\dot{x}_{ij} = -x_{ij} + \alpha y_{ij} + A\left[\sum_{\substack{h=1,\dots,N\\ h\neq j}} y_{ih} + \sum_{\substack{k=1,\dots,M\\ k\neq i}} y_{kj}\right] + I + \beta w_{ij} + B\left[\sum_{\substack{h=1,\dots,N\\ h\neq j}} w_{ih} + \sum_{\substack{k=1,\dots,M\\ k\neq i}} w_{kj}\right] \qquad (5)$$

The neighborhood effect is considered with the terms in square brackets.

The output equation:

$$y_{ij} = g(x_{ij}) = \frac{1}{2}(|x_{ij}+1| - |x_{ij}-1|), \qquad 1 \leq i \leq M; \quad 1 \leq j \leq N, \qquad (6)$$

convert the state of each cell into a binary output, where $y_{ij} = 1$ means that one packet from queue (i, j) can be transmitted. For each logical queue at the beginning of a given time slot two terms are fed to each cell: the initial state $x_{ij}(0)$ of the queue, looking at the presence of waiting packets, and the input term, w_{ij}, that is held constant for the whole time slot, related to the priority of packets in each logical queue. Therefore we define two $M \times N$ matrices for the entire system: one containing initial states (S) and one containing the input terms (W). A value $s_{ij} = 1$ means that there is at least one packet in that queue waiting to be commutated, and it is used to initialize the output values ($y_{ij}(0) = g(x_{ij}(0)) = s_{ij}$). Input values are calculated through nonlinear functions of the traffic class, the number of waiting packets and the time spent inside the queue (Fig. 5).

In DiffServ approach a switching unit, that is a core node, do not handle separately each application flow; the edge nodes aggregate applications packets with similar QoS requirement (PHB group) in the same flow. In this paper three PHB groups has been considered: Expedited Forwarding (EF), Assured Forwarding (AF) and Best Effort (BE).

The EF class can be used to build a low loss, low latency, low jitter, assured bandwidth, end-to-end service through DiffServ domains, in order to overcome loss, latency and jitter introduced by queues in each node. The priority function has been implemented as shown in Fig. 5(a), where the priority reach the maximum value whenever one packet is inside the queue.

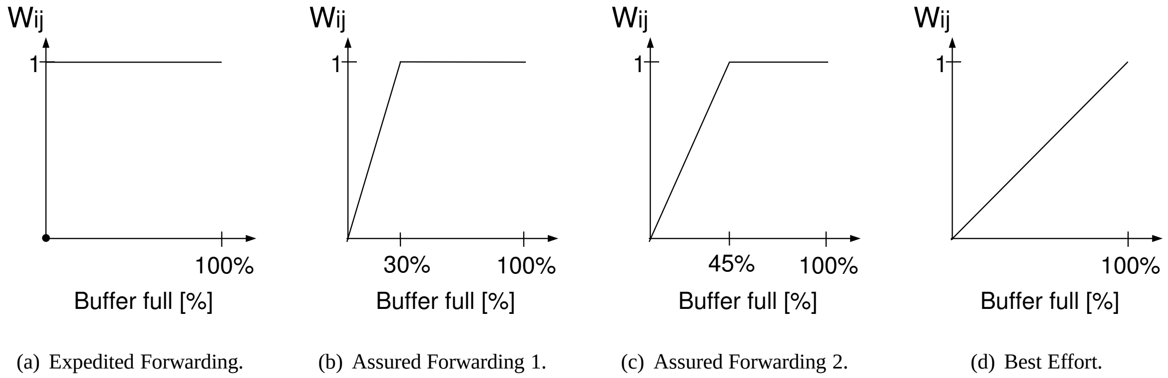
The AF class has been introduced in order to provide inside a DiffServ domain different levels of forwarding assurances; in particular four level of emission priorities are defined, and for each class, in each DiffServ node, a certain amount of forwarding resources (buffer space and bandwidth) is allocated. Within each AF class three possible dropping functions have been implemented, taking into consideration the priority of each packet. Two classes of Assured Forward have been considered in this paper, using two non linear function (Figs. 5(b) and 5(c)) with different *saturation levels* that arrive at the maximum value for a buffer filling of 30% and 45% respectively.

The BE class has been introduced for traffic without particular QoS requirements in terms of delay, jitter and packet loss but, despite this, a linear function growing with the number of packets in the queue has been used (Fig. 5(d)). The total waiting time has been introduced in order to solve the starvation problem for the low priority traffic class (aging method); otherwise, in case of high priority heavy traffic, low class packets could remain in starvation. This additional term is present in both AFs and BE input values computation.

In Tab. 1, different requirements in terms of bandwidth and different sensitivity to delay, jitter and packet loss for typical multimedia application are shown.

Through input values all QoS requirements for each class have been introduced: higher are w_{ij} values for the cell $C(i,j)$, and higher is the priority of the corresponding queue.

During the design phase, the goodness of each solution has been evaluated through a cost and an efficiency (η)



(a) Expedited Forwarding. (b) Assured Forwarding 1. (c) Assured Forwarding 2. (d) Best Effort.

Figure 5: w_{ij} evaluation for each PHB

Table 1: Application Performance Dimension

	Performance dimension			
		Sensitivity to		
Application	**Bandwidth**	**Delay**	**Jitter**	**Loss**
VoIP	Low	High	High	Med
Video Conferencing	High	High	High	Med
Streaming video	High	Med	Med	Med
Streaming audio	Low	Med	Med	Med
eBusiness (Web browsing)	Med	Med	Low	High
E-mail	Low	Low	Low	High
File Transfer	Med	Low	Low	High

functional defined as:

$$ \tag{7} $$

$$ Z(X,W) = \sum_{\substack{i=1,\dots,M \\ j=1,\dots,N}} x_{ij} w_{ij} \tag{8} $$

$$ \tag{9} $$

$$ \eta = 100 \left[1 - \frac{B_P}{T_P} \right] \% \tag{10} $$

$$ \tag{11} $$

where x_{ij} is the output of the cell $C(i,j)$ with a cost $Z(X,W)$, B_P is the number of valid solutions X' with a cost $Z(X',W)$ higher than $Z(X,W)$, T_P is the total number of valid solutions that do not violate imposed constraint. Using this definition, the parameters of (5) have been chosen as $a = 2$, $A = -40$, $b = 20$, $B = 1$, $I = 1$ in order to guarantee convergence and maximize efficiency. In Fig. 6 the evolution of the state of each cell of a CNN is shown for a satellite with 4 inputs, 4 outputs and 3 traffic classes. It is possible to see that only 4 neurons reach a state value higher than 1 or equivalent an output equal to 1. All the other cells have a final state lower that 0, resulting in a no-transmission state.

5. NUMERICAL RESULTS

Numerical results have been performed in order to estimate delay, jitter and packet loss for each PHB in different traffic conditions and considering different system dimensions changing from 4 to 128 input/output beams. The input traffic model is a Geometric Modified with a message arrival probability equal to $(1-p)$:

$$ \text{Prob}\{x = k\} = p^{k-1}(1-p), \qquad 0 \le p \le 1; \quad k = 1, 2, \dots \tag{12} $$

where (12) represents the probability to have a message after $k-1$ empty cells. Each message is made of packets with the same destination and priority, belonging to the same traffic aggregate. The message length distribution follows a heavy-

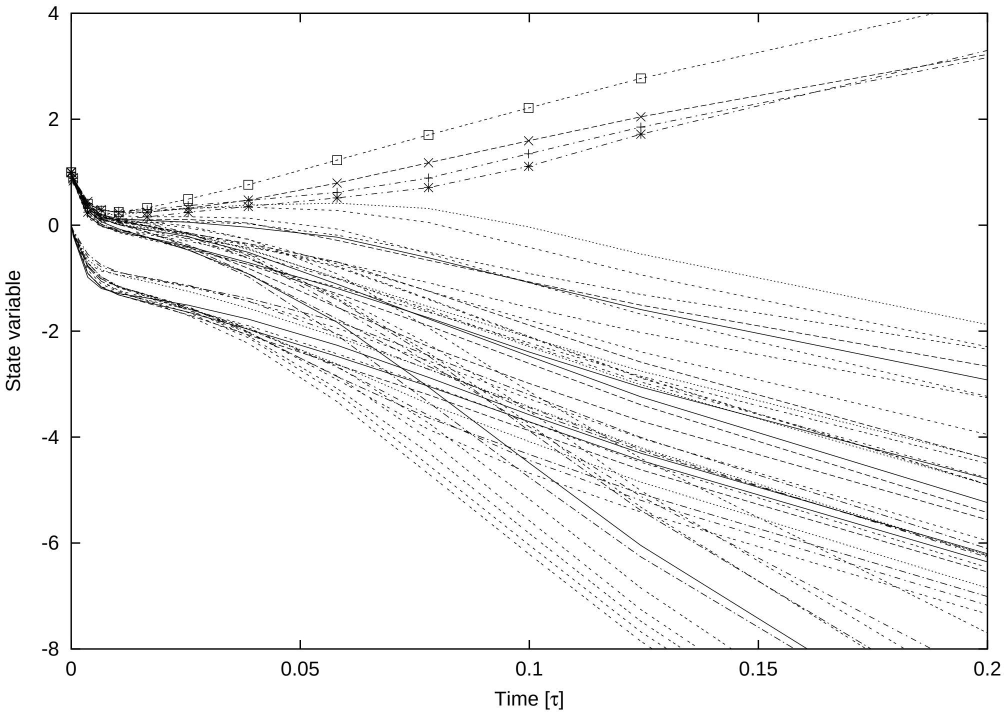


Figure 6: State progress.

tailed model; the messages length has a variable number of packets in accordance with the trunked Pareto distribution:

$$f(x) = \begin{cases} \frac{\alpha k^{\alpha}}{x^{\alpha+1}}, & k \leq x < m \\ \frac{k^{\alpha}}{x^{\alpha}}, & x = m \\ 0, \quad x < k, & x > m \end{cases} \tag{13}$$

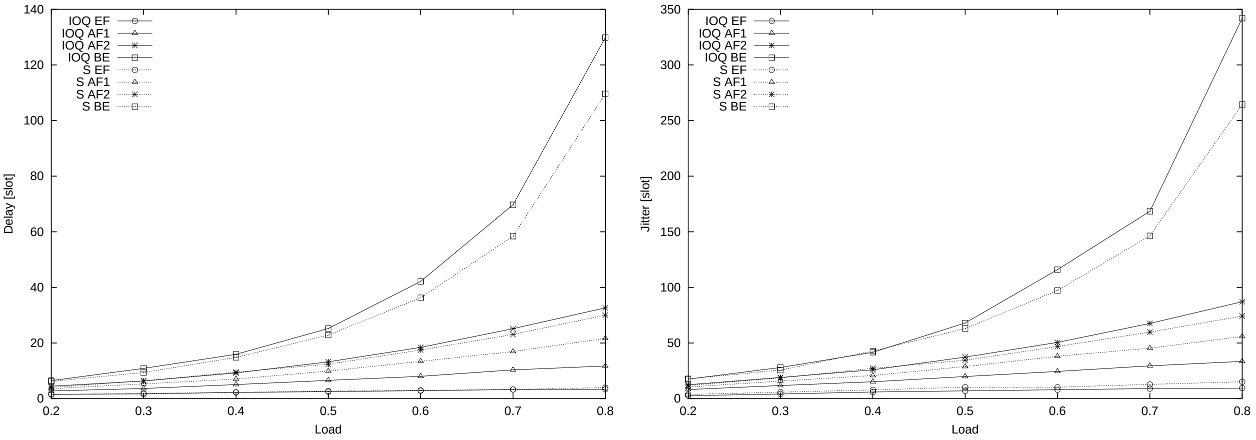
where k and m are the minimum and maximum dimension (in packet) of the message while α is a characteristic parameter, called *shape*, equal to 1.1 [11]. This distribution is a good model for typical bursty Internet traffic.

The delay analysis have been performed measuring the time spent in the queue by each packet. The jitter is based on difference between delays of two consecutive packets and it expresses the variability of the delay itself.

The presented delay and jitter tests are performed with a fixed system dimension equal to 16 inputs, 16 outputs, 4 traffic classes and total traffic load increasing from 0.2 to 0.8; the results, shown in Fig. 7, are compared to the results obtained using a Multiple Input Queuing with Shared Input Buffer (IOQ). It is possible to see that the delay behavior of class EF and BE is very similar to IOQ; for AF classes the effects of the aging and the *dynamic priority* is highlighted (Fig. 7).

A satellite switch needs a careful analysis of buffers dimension and position; considering a 16 inputs/outputs system able to manage 4 traffic classes three different approaches have been considered: 16 shared input buffers, 16×4 shared input buffers derived from previous approach, dividing each queue in priority and one common buffer. Fixing the dimension of the system and the number of traffic classes, the total memory allocated and its position have been changed (Tab. 2). Buffer occupancy analysis has been also performed, considering an infinite buffer dimension; in Fig. 8(a) buffer occupancies probability in packets for each PHB are shown. For several system dimensions, as shown in Tab. 2, it has been also measured (Fig. 8(b)) the mean packet loss probability for the Common buffer (CB), the Shared input buffer (SB) and the Shared input and priority buffer (PB) approach. In particular, a switch with a higher buffer capability permits to have lower packet loss, respecting QoS classes order. The best solution seems to be the Shared Input, resulting to have the best trade-off between packet loss and reading speed.

Similar results have been also achieved for simpler traffic type; we have omitted related figures for sake of brevity.



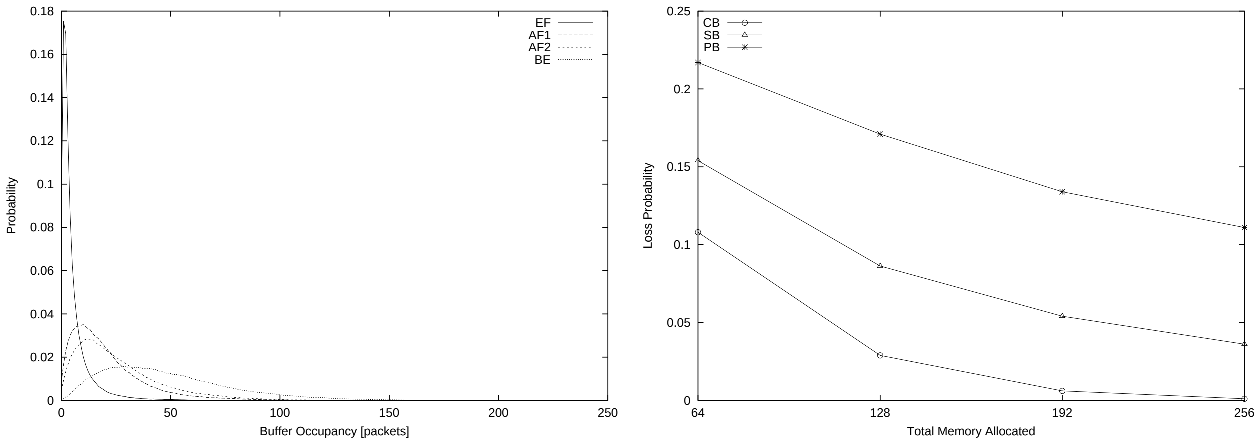
(a) Delay.

(b) Jitter.

Figure 7: Delay and jitter.

Table 2: Memory allocation

		Buffer dimension (packets)					
Allocation	Number of buffers in the system	Case 1	Case 2	Case 3	Case 4	Buffer Reading Speed (CNN)	Buffer Reading Speed (IOQ)
Shared input and priority	64	1	2	3	4	1	16
Shared input	16	4	8	12	16	4	16
Common	1	64	128	192	256	16	16



(a) Buffer Occupancy.

(b) Packet loss probability.

Figure 8: Buffer Occupancy and Packet loss probability.

6. CONCLUSIONS

In this paper a switch for satellite systems has been proposed. Actually, Internet traffic is managed with Best Effort policy; on the other hand multimedia services have gained more importance, requiring a more efficient approach in resource management. DiffServ seems to be the most attractive technique in order to guarantee certain QoS levels; for satellites systems the introduction of DiffServ techniques became crucial because of the reduced computational capabilities required on-board.

In this paper a switch that use a Cellular Neural Network as an arbiter in order to choose the packets to be delivered to each output beam is proposed, respecting some QoS and physical constraints. Several QoS classes have been implemented with different parameters in terms of delay, jitter and loss. The choose of the packets to be delivered is made maximizing a cost function. In particular the proposed solution is easily adaptable to various environment and traffic type, taking into account that a change in priority policy could be made modifying only the related nonlinear functions.

Numerical results have been driven with a heavy-tailed traffic model. The goodness of the proposed switch in terms of delay and jitter is then shown.

REFERENCES

[1] E. Del Re, L. S. Ronga, and L. Pierucci. Trends in satellite communications. In *Proc. of IEEE ICC 2002*, volume 5, pages 2938–2988, New York, NY, USA, April 2002.

[2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss. An Architecture for Differentiated Services. RFC 2475, December 1998.

[3] E. Del Re and R. Fantacci. An efficient high-speed packet switching with shared input buffers. In *Proc. of GLOBECOM '92*, volume 3, pages 1472–1476, Orlando, FL, USA, December 1992.

[4] H. Ahmadi and W. E. Denzel. A Survey of Modern High-Performance Switching Techniques. *IEEE Journal on Selected Areas in Communications*, 7(7):1091–1103, September 1989.

[5] M. J. Karol, M. G. Hluchyj, and S. P. Morgan. Input versus output queueing on space-division packet switch. *IEEE Transactions on Communications*, 35(12):28–34, December 1987.

[6] L. O. Chua and L. Yang. Cellular neural networks: Theory. *IEEE Transactions on Circuits and Systems*, 35(10):1257–1272, October 1988.

[7] L. O. Chua and L. Yang. Cellular neural networks: Applications. *IEEE Transactions on Circuits and Systems*, 35(10):1273–1290, October 1988.

[8] J. J. Hopfield. Neurons with graded response have collective computational properties like those of two state neurons. *Proc. Nat. Acad. Sci.*, 81:3088–3092, 1984.

[9] V. Jacobson, K. Nichols, and K. Poduri. An Expedited Forward PHB. RFC 2598, June 1999.

[10] J. Heinanen, F. Baker, W. Weiss, and J. Wroclawski. Assured Forward PHB Group. RFC 2597, June 1999.

[11] Qos concept and architecture. Technical Specification 25.107, 3GPP, 06 2000.