Reprint

Performance Analysis of the Synchronization Mechanism used at the VIRUS Interface

A. Maniatopoulos, T. Antonakopoulos and V. Makios

The 3rd IEEE International Conference on Electronics, Circuits and Systems

RHODES, GREECE, OCTOBER 1996

PERFORMANCE ANALYSIS OF THE SYNCHRONIZATION MECHANISM USED AT THE VIRUS INTERFACE

A. Maniatopoulos, T. Antonakopoulos and V. Makios

Division of Telecommunications and Technology of Information
Department of Electrical and Computer Engineering
University of Patras, 26500 Patras, Greece

Tel: +30 (61) 997 286, Fax: +30 (61) 997 342, e-mail: antonako@ee.upatras.gr

Abstract

In this paper the performance analysis of a new high-speed cell-based interface is presented. The so called VIRUS interface can be used for replacing existing physical layer interfaces of cell-based networks, since its functions result to lower hardware complexity and thus, it can be easily used in higher data rates. The interface analysis is mainly focused on the evaluation of the proposed synchronization method as well as on the effects of the line coding scheme on the synchronization procedure. The results prove the advantage of the proposed interface when used for cell-based communication networks and high-speed point-to-point interfaces.

I. INTRODUCTION

Networks using asynchronous mode of transfer are generally characterized by their ability to retain maximum throughput even when the system is overloaded. The Fibre Channel (FC) and Asynchronous Transfer Mode (ATM) belong to this category of protocols which, among other similarities, are independent of their physical rate, allowing for scalability to higher speeds in the future [1]. ATM and FC support the simultaneous transmission of different information types and are used in integrated services networks.

The interface presented in this paper has been based on the advantages of both these transfer modes and provides the functionality required by the ATM Layer. Its name VIRUS comes out from the generated sequences which are called Idle Order Sets (IOS), since their acronym IOS is the translation of the word '***virus***' in the Greek language.

Section II highlights the basic functions of the VIRUS interface, introducing the notion of the pseudo-frame and emphasizing on the common functions among the VIRUS, ATM and FC protocols. The analytic formulas of the VIRUS synchronization parameters are derived in section III, using an imbedded Markov chain, while section IV describes how the interface parameters influence its performance.

II. BASIC PRINCIPLES AND FUNCTIONAL DESCRIPTION.

The IOS is a special type of packet whose length is much smaller than that of an ATM cell. This type of packet consists of eight characters encoded using the 4B1C code [2] and some violations in order to provide idle sequences similar to that of the FC which uses the 8B/10B code [3]. User cells (UC) are also encoded using the 4B1C code but without violations and this guarantees the uniqueness of the IOS pattern in the serial bit stream The beginning of the synchronization procedure is based on the detection of an IOS and the transmitter is responsible for inserting IOSs within the cell stream every k-consecutive which guarantees that an FC receiver port, being in the loss-of-synchronization state, will establish word boundaries in no more than k-cells period [4]. In VIRUS interface the sequence of an IOS succeeded by λ-cells ($0 \leq \lambda \leq k$), is called VIRUS pseudo-frame (VPF).

Fig. 1 shows the VIRUS interface state diagram for the transmit direction. It includes a T_X-state counter, c_H, which indicates the number of user cells transmitted after the last IOS. Initially the c_H is equal to zero. When a user cell is transmitted, the counter increases. When it reaches its maximum value k, the user cell buffer service is ceased, a pair of IOSs is transmitted and the T_X-state counter is cleared.

At the receive side, the IOSs are detected and removed by the TC sub-layer where the cell delineation mechanism is implemented, based on the

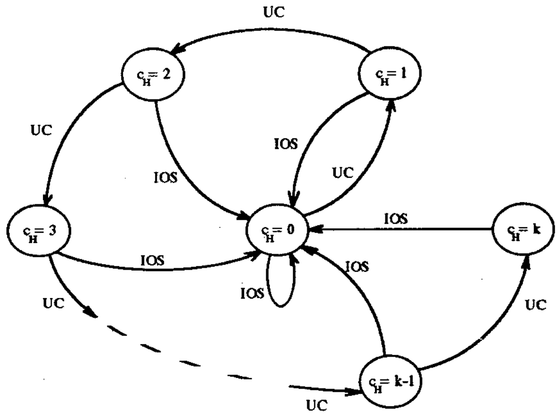

Fig. 1. The state diagram of the transmitter of the VIRUS interface

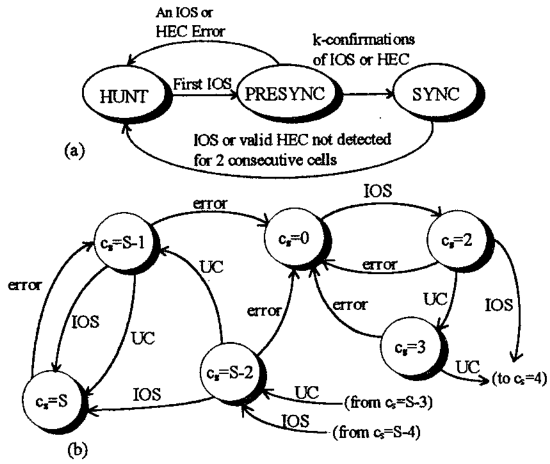

Fig. 2 (a) The cell delineation state diagram (b) The Rx-state counter operation

'HEC validation' method [5] and on the 'Indication of IOS' [4].

The VIRUS receiver synchronization diagram is shown in Fig. 2a. Initially the synchronizer of the cell delineation mechanism is in the HUNT state and changes to the PRESYNC state only when a pair of valid IOSs are detected by the boundaries detector [4]. In this state, two validation modules are triggered, the HEC module and the IOS module due to the difference in the length between the User Cells (UC) and the IOSs. If any valid indication is available at the end of the eighth character, a new cell or an IOS boundary is determined. The synchronizer uses the Rx-state counter (c_S) which increases its value by one or two, whether a correct UC header or a valid IOS is detected respectively (Fig. 2b).

III. ANALYSIS OF SYNCHRONIZATION TIME

For the evaluation of the VIRUS synchronization method, the mean value of the synchronization time has been studied. For the rest of this paper, the following terminology is used:

T_{UC}, T_{IOS} : transmission times of a UC, IOS respectively,

λ_{UC} : the rate of UC arrivals,

λ_{IOS} : the rate of IOS generations,

$\bar{t}_{HUNT}$, $\bar{t}_{PRESYNC}$: the mean value of the time the system spends in the HUNT and PRESYNC states respectively,

P_j : the probability the transmitter is in the j-state at a given instant,

p_e : the probability the UC-buffer is empty,

$\bar{t}_{SAT}$: the mean value of synchronization acquisition time,

m_{UC}-m_{IOS} : the length in characters of a UC-IOS,

k : the maximum number of user cells that can be transmitted between two IOSs,

c_H : the value of the Tx-state counter,

c_S : the value of the Rx-state counter,

n_i : the number of cells in the UC-queue at the i[th] departure epoch,

$\bar{t}_{i,IOS}$: the mean value of the time needed for the transmission of the next IOS when the system is in the i-state,

The following analysis has been based on the model shown in Fig. 3a. The UC-queue has priority over the IOS queue as long as the Tx-state counter value, c_H, is lower than its maximum value. The model uses non-preemptive priorities, which are not constant but toggle, depending on the Tx-state

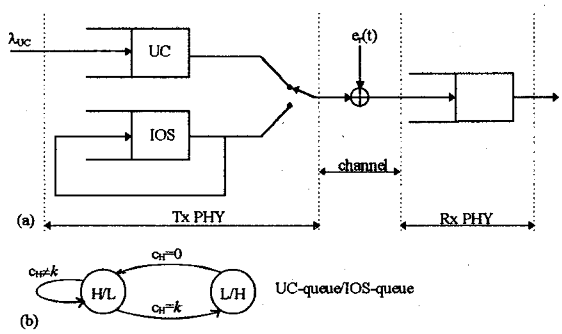

Fig. 3 (a) The model of the VIRUS interface and (b) its priorities

counter value. Whenever the Tx-state counter gets its maximum value k, the IOS-queue has the highest priority for transmitting the next cell. On the other hand, no message interruption is allowed during transmission. The system analysis relies on the properties of the imbedded Markov chain [7]. It is assumed that the UCs arrive in the UC-queue independently from the system state and they follow a Poisson process, with mean value equal to λ_{UC} and their length is constant and equal to m_{UC}. The model description can be based on two state equations. The first state equation is valid whenever there are cells in the UC-queue:

$$\begin{aligned} n_{i+1} &= n_i - 1 + \alpha_1 + (1 - \alpha_1 + \alpha_2) . U[c_i - k] \quad n_i > 0 \\ c_{i+1} &= (c_i + 1).(1 - U[c_i - k]) \end{aligned} \tag{1}$$

where $U[x]=1$ for $x \geq 0$ and 0 for $x < 0$, c_i is the value of the T_x-state counter and α_1 is the number of UCs which arrive during an UC transmission. If the UC-queue is empty, then:

$$\begin{aligned} n_{i+1} &= \alpha_2 \qquad n_i = 0 \quad \text{and} \quad \forall c_i \\ c_{i+1} &= 0 \end{aligned} \tag{2}$$

where α_2 is the number of UCs which arrive during an IOS cell transmission. By using the consideration of [7] according to a priority system, if Π_0 is the probability of the system being empty:

$$\Pi_0 \equiv P[n_i = 0, n_{i,IOS} = 0] = 1 - \lambda_{UC} m_{UC} - \lambda_{IOS} m_{IOS} \tag{3}$$

where $p \equiv \lambda_{UC} m_{UC} + \lambda_{IOS} m_{IOS}$ is the traffic intensity. Since the IOS-buffer always contains an IOS, $n_{i,IOS} = 1$ and $\Pi_0 = 0$ (4)

thus $\quad \lambda_{UC} m_{UC} + \lambda_{IOS} m_{IOS} = 1$ (5)

Based again on [7], on the consideration that over a long time interval the probability of a q-epoch is equal to the probability of a class q message arriving to a non-empty queue, thus:

$$p_e = P\left[n_i = 0\right] = \frac{\lambda_{IOS}}{\lambda} \tag{6}$$

Solving equations (5) and (6), one gets:

$$p_e = \frac{1 - \lambda_{UC} m_{UC}}{1 - \lambda_{UC}(m_{UC} - m_{IOS})} \tag{7}$$

Since the transmission of an IOS forces the system to the 0-state, the system goes to the next state only when there are user cells in the UC-buffer. Thus, the system equilibrium equations become:

$$\begin{aligned} P_i &= P_{i-1} \cdot (1 - p_e) \qquad 1 \leq i \leq k \\ P_o &= P_o p_e + \sum_{i=1}^{k-1} P_i p_e + P_k \end{aligned} \tag{8}$$

while the system normalizing condition is:

$$\sum_{i=0}^{k} P_i = 1 \tag{9}$$

Solving the system of equations (8) and (9), one gets:

$$P_i = \frac{p_e \cdot (1 - p_e)^i}{1 - (1 - p_e)^{k+1}} \tag{10}$$

The mean value of synchronization acquisition time, is given by:

$$\bar{t}_{SAT} = \bar{t}_{HUNT} + \bar{t}_{PRESYNC} \tag{11}$$

By definition, the mean value of the HUNT-time is equal to the mean value of inter-IOS times, that is:

$$\bar{t}_{HUNT} = \sum_{i=0}^{k} P_i . \bar{t}_{i,IOS} + \frac{\lambda_{UC}}{\lambda}(T_{UC} - T_{IOS}) \tag{12}$$

If $\lambda \equiv \lambda_{UC} + \lambda_{IOS}$ is the total message arrival rate of either UCs or IOSs, then the probability of a message being a UC is equal to $\frac{\lambda_{UC}}{\lambda}$. The $\bar{t}_{i,IOS}$ is derived from the state diagram of Fig. 2b:

$$\bar{t}_{i,IOS} = T_{IOS} + T_{UC} \cdot (1 - p_e) \cdot \frac{1 - (1 - p_e)^{k-i}}{p_e} \tag{13}$$

thus:

$$\bar{t}_{HUNT} = T_{IOS} + \frac{\lambda_{UC}}{\lambda} \cdot (T_{UC} - T_{IOS}) + \frac{(1 - p_e)\left[1 - (1 - p_e)^{k+1} - (k+1) p_e (1 - p_e)^k\right]}{p_e\left[1 - (1 - p_e)^{k+1}\right]} T_{UC} \tag{14}$$

For calculating the mean value of the time the system stays in the PRESYNC state, $\bar{t}_{PRESYNC}$, we also follow the methodology used for $\bar{t}_{HUNT}$ and we conclude that:

$$\begin{aligned} \bar{t}_{PRESYNC} = &\sum_{i=0}^{\left\lfloor \frac{k}{2} \right\rfloor} \frac{(k-i)!}{i!(k-2i)!} p_e^{\,i} (1 - p_e)^{(k-2i)} \left[i \cdot T_{IOS} + (k - 2i) T_{UC}\right] \\ &+ \sum_{i=1}^{\left\lceil \frac{k}{2} \right\rceil} \frac{(k-i)!}{(i-1)!(k-2i+1)!} p_e^{\,i} (1 - p_e)^{(k+1-2i)} \cdot \left[i \cdot T_{IOS} + (k + 1 - 2i) T_{UC}\right] \end{aligned} \tag{15}$$

The section that follows shows how the system parameters (like k, s values) influence the total time required for synchronization.

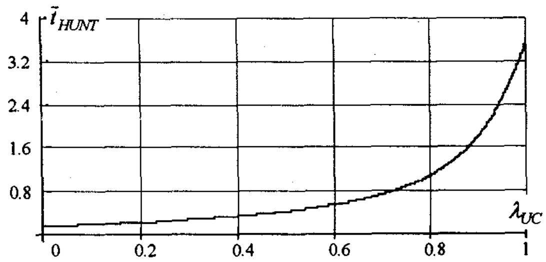

Fig. 4 Mean value of the time the system spends in HUNT state (*k*=5)

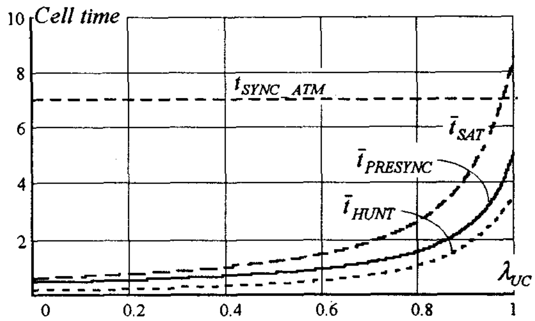

Fig. 5 The mean time values of the sync states (k=5)

IV. NUMERICAL RESULTS

Unlike to the VIRUS method, the synchronization time in the ATM standard has a fixed value and is equal to seven times the cell duration. Considering the limit of $\bar{t}_{HUNT}$ and $\bar{t}_{PRESYNC}$ when λ_{UC} tends to 1, it can be derived that under heavy traffic conditions the maximum value of $\bar{t}_{SAT}$ is approximately equal to T_{IOS}+(k+ k/2+1)T_{UC}. In Fig. 4 the $\bar{t}_{HUNT}$ versus λ_{UC} timing diagram is illustrated, while Fig. 5 shows how the offered load influences the total synchronization time and its components. The dependence of the synchronization time components to the cell delineation parameters is shown in Fig. 6.

The results that are derived from the above numerical analysis, show that appropriate selection of system parameters leads to the minimization of the time spent by the system in the loss-of-synchronization state. In normal cases, the VIRUS synchronization method becomes much faster than that of the ATM interfaces, which makes it suitable for Customer Premises Network (CPN) applications.

V. CONCLUSIONS

In this paper, a new interface called VIRUS for use in cell-based systems was presented. The main

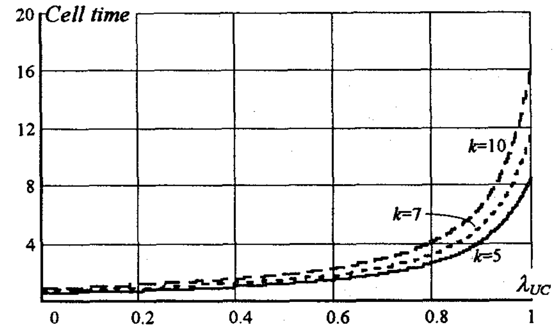

Fig. 6 Influence of *k*-system parameter in the synchronization acquisition time

advantage of the proposed interface is its implementability in high speed links. An implementation of the VIRUS interface at using commercial available components and FPGAs has been presented in [5].

REFERENCES

[1]. Jan Dedek, "Choosing between Fibre Channel and ATM", EDN, February 1996

[2]. N. Yoshikai, K. Katagiri and T. Ito: "mB1C Code and Its Performance in an Optical Communication System", *IEEE Transactions on Communications,* Vol. Com-32, No. 2, February 1984, pp. 163-168.

[3]. "Fibre Channel - Physical and Signalling Interface (FC-PH) Rev. 3.0", ANSI X3T9 I/O Interface, Working Draft, June 1992.

[4]. A. Maniatopoulos, T. Antonakopoulos and V. Makios: "Synchronization Issues of a Customer Premises Network Based on Cell Transmissions over Fibre Channel Interfaces", *Proceedings of the 20th EUROMICRO Conference,* Liverpool, September 1994.

[5]. CCITT Recommendation I.432: "B-ISDN User-Network Interface - Physical Layer specifications", June 1992.

[6]. Martin de Prycker: "Asynchronous Transfer Mode: Solution for Broadband ISDN", Ellis Horwood Ltd, 1992.

[7]. J. F. Hayes: "Modeling and Analysis of Computer Communication Networks", Plenum Press, 1986