Virtual Token-Passing Ethernet Proposal using Programmable Hardware

Francisco Borges Carreiro\textsuperscript{1,2}, José Alberto Fonseca\textsuperscript{2}, Francisco Vasques\textsuperscript{3}

\textsuperscript{1}Departamento de Eletro-Eletrônica – CFET-MA
65025-001 - São Luís Maranhão - Brasil

\textsuperscript{2}Departamento de Electrónica e Telecomunicações – Universidade de Aveiro/IEETA
3810143 - Aveiro - Portugal

\textsuperscript{3} DEMEGI - Faculdade de Engenharia UP
4200-465 Porto – Portugal

fborges@ieeta.pt, jaf@det.ua.pt, vasques@fe.up.pt

Abstract. The Virtual Token-passing Ethernet (VTPE) is a real-time shared Ethernet approach based on an implicit token rotation principle known as virtual token-passing. The first VTPE version has been implemented in an 8 bit microcontroller using a standard Ethernet controller. The use of low processing power microcontroller and standard Ethernet controllers, despite of the low processing VTPE requirements, imposes limitations in the network efficiency throughput. This paper proposes a new implementation using programmable hardware (FPGA) having a physical transceiver as media interface, instead of a standard MAC controller. This proposal increases the throughput to the theoretical limits of Ethernet 10/100 Mbps.

1. Introduction

The first implementation of VTPE was aimed to be used in small processing power microcontrollers and standard Ethernet controllers. In the first implementation the network efficiency throughput in an Ethernet 10 Mbps network, using a PIC 18F458 microcontroller at 40MHz clock, was demonstrated to be 17.5% [Carreiro et al., 2004]. This low efficiency throughput is because VTPE runs outside the Ethernet controller and due to the low processing power of the microcontroller as well.

The new VTPE implementation, VTPE-MAC, arbitrates the bus during a frame passing and the inter-frame gap (IFG), so no extra arbitration time is wasted. Consequently, the saved time is reverted in the efficiency throughput. This improvement is possible due to the low VTPE processing requirements, to the processing power of FPGAs and to the use of the transceiver, which makes possible to run VTPE during the frame transmission.

This new implementation will permit to reach, deterministically, the theoretical limits of Ethernet 10/100Mbps efficiency throughput (54.6% for minimal size frame and 97.5% for maximum size frame) that is found when there is a single transmitting node in the bus. For a shared Ethernet segment with more than one node these throughput values are unreachable due to the collisions and the probabilistic resolution algorithm (backoff).
The remaining of this paper is: section 2 presents a review of VTPE. Section 3 presents the VTPE-MAC proposal and section 4 the conclusions and future work.

2. VTPE Overview

A VTPE system consists of a shared Ethernet bus with the producer nodes disposed in a logical ring [EN 50170], [Carreiro et al., 2003]. Each producer node has a node address (NA), between 1 and the number of producers expected within a system. All producers have an Access Counter (AC) which identifies the node that can access the bus in a specific time interval. To implement a virtual token-passing schema using Ethernet, Ethernet’s broadcast destination address must be used because all devices must read each frame dispatched on the bus. Whenever a frame is sent to the bus, an interrupt must be generated in all producer nodes. After the interrupt, all nodes increase their ACs and the producer node whose AC value is equal to its own unique address is allowed to access the bus. If the actual node doesn’t have anything to transmit (or indeed is not present) the bus becomes idle and, after a certain time, all the access counters are increased by one. The next producer is then allowed to access the bus. If, again, it has nothing to transmit, the bus continues idle and the described procedure is repeated until a producer effectively uses the bus. When the access counter exceeds the number of producers, it is preset to 1 and the cycle is repeated again.

All producers have a timer, which can be programmed with time value $t_1$ or $t_2$. The time $t_1$ is the VTPE inter-frame gap and must be enough to attend the interrupt of the transmitted packet, decode the transmitted frame (check if one or more messages encapsulated in the Ethernet data field are designated to the node), run the access counters, and transmit if the node has any ready frame to transmit. The time $t_2$ is a guard time needed to detect nodes absent from the network or that, despite being present, don’t have anything to transmit guarantying that the next node gets the bus.

2.1 Parameters $t_1$, $t_2$ and real-time analysis

For implementation in programmable hardware, the frame is checked during its passage, so $t_1$ can be shortened as much as the Ethernet IFG (9.6µS for 10Mbps and 0.96µS for 100Mbps). Then this value permits reaching the maximum Ethernet throughput. With regard to $t_2$ a value around 64 bit times (preamble and SFD) should be adequate, i.e. 6.4µS for 10Mbps and 0.64µS for 100Mbps.

For real-time analyses, let’s consider a transmitting scenario in a VTPE system as shown in figure 1. As it is shown there, each node transmits a single frame per token holding time, starting at node 1. After node 1 transmission, node 2 gets the right of transmission, and so on up to the last node, the node $n$. After node’s $n$ transmission, node 1 gets the right of transmission again.

![Figure 1. VTPE-MAC token rotation time scenario](image)
The $T_{RT}$ is the time between two consecutive instants in which a specific node gets the right to transmit. The $T_{RT}$ can be found based on the scenario depicted in the figure 1 and using the equation (Eq.1), where $n$ is the number of nodes, $t_1$ is like mentioned before, and $t_{pk}$ is the time to transmit a VTPE packet from node $k$.

The highest TRT, token rotation time, occurs when all nodes transmit their maximum packets. So the maximum Token Rotation Time $\text{max}T_{RT}$ can be calculated by equation (Eq.2).

$$T_{RT} = n * t_1 + \sum_{k=1}^{n} t_{pk} \quad \text{(Eq.1)}$$

$$\text{max}T_{RT} = n * t_1 + \sum_{k=1}^{n} \text{max}(t_{pk}) \quad \text{(Eq.2)}$$

The extension of the $T_{RT}$ analysis for VTPE-MAC is direct because the only timeliness modification implied by the use of the VTPE-MAC is the shortening of $t_1$ and $t_2$. Then the equation Eq.1, for a generic $T_{RT}$ case and Eq.2 for worst-case $T_{RT}$ at 10Mbps can be rewritten as:

$$T_{rt} = 9.6 * n + \sum_{k=1}^{n} t_{pk} \mu S \quad \text{(Eq.3)}$$

$$\text{max}T_{rt} = 9.6 * n + \sum_{k=1}^{n} \text{max}(t_{pk}) \mu S \quad \text{(Eq.4)}$$

In the same way the Eq1.and Eq.2 for 100Mbps can be rewritten as:

$$T_{rt} = 0.96 * n + \sum_{k=1}^{n} t_{pk} \mu S \quad \text{(Eq.5)}$$

$$\text{max}T_{rt} = 0.96 * n + \sum_{k=1}^{n} \text{max}(t_{pk}) \mu S \quad \text{(Eq.6)}$$

3. VTPE-MAC Proposal

In this proposal, a VTPE node is composed of an Ethernet transceiver and its accessory parts (magnetic transformers and RJ45 connector), a FPGA where the VTPE-MAC firmware is implemented and a processor/microcontroller where the application runs.

This section focuses in the VTPE-MAC proposal, so the Ethernet transceiver and the microcontroller/processor aren’t presented.

3.1 VTPE-MAC block

A simplified block diagram of the VTPE-MAC core is shown in figure 5. The Receive Control Block controls the frame’s reception from the Ethernet transceiver and delivers the received frame to the VTPE Frame Decoder and signalises to the AC Counter and NA=AC Checker that a frame was received.

The VTPE frame decoder checks the Source Address to actualise the node’s active table, the Data/Type field, and the data field. If there is relevant data (VTPE messages) it delivers them to the memory, otherwise it discards the frame.

The AC and Checker block increments the AC and compares its value with NA. It sets a flag to “1” if AC=NA or to “0” if AC≠ NA.

The Transmission Control Block gets a frame to be transmitted from the VTPE Frame Generator, controls the $t_1$ and $t_2$ timers, and controls the frame transmission.

The VTPE Frame Generator gets data from the Memory (VTPE messages), generates the CRC, adds padding bits to complete the 46 minimum bytes, if necessary, and encapsulates all this data in the Ethernet frame and delivers it to the Transmission Control Block so that it can be transmitted to the bus.
The memory consists of two memory blocks in ring format: The Received Data Ring and the Transmit Data Ring. Two local DMA (Direct Memory Access Controller) channels, not shown in figure 2, are used to manage received data and to manage the transmission of data. The first one, during a packet reception, stores the received data from the VTPE Frame Decoder into the Received Data Ring and, during a frame transmission, transfers data from the Transmit Data Ring to the VTPE Frame Generator to be transmitted to the controller. The second DMA channel is used to transfer data between memory (Received Data Ring or Transmit Data Ring) and the host processor.

This proposal focuses in the MAC controller architecture, so the memory management and the bus interface between the host and the VTPE-MAC will be specified in the future.

4. Conclusions

A new implementation of VTPE was proposed. Its throughput improvement, real-time characteristics, and block diagram were also discussed. This implementation will permit deterministically to reach the maximum throughput of Ethernet because $t_1$ (VTPE inter-frame gap) is shortened to the Ethernet inter-frame gap value.

References

EN 50170, Volume 1- European Fieldbus Standard
