

PONTIFÍCIA UNIVERSIDADE CATÓLICA
DO RIO DE JANEIRO



Fernando Diniz Hämmerli

**Desenvolvimento de um Analisador de Elementos
de Rede Baseado no Padrão Gigabit Ethernet**

Dissertação de Mestrado

Dissertação apresentada como requisito parcial para
obtenção do título de Mestre pelo Programa de Pós-
Graduação em Engenharia Elétrica da PUC-Rio.

Orientador: Marbey Manhães Mosso

Rio de Janeiro

Julho de 2008

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.



Fernando Diniz Hämmerli

Desenvolvimento de um Analisador de Elementos de Rede Baseado no Padrão Gigabit Ethernet

Dissertação apresentada como requisito parcial para obtenção do título de Mestre pelo Programa de Pós-Graduação em Engenharia Elétrica da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

Prof. Marbey Manhães Mosso

Orientador

Centro de Estudos de Telecomunicações - PUC-Rio

Prof. Rodolfo Saboia Lima de Souza

Centro de Estudos de Telecomunicações - PUC-Rio

Prof. Marco A. Grivet Mattoso Maia

Centro de Estudos de Telecomunicações - PUC-Rio

Gláucio Lima Siqueira

Centro de Estudos de Telecomunicações - PUC-Rio

Prof. José Eugênio Leal

Coordenador Setorial do Centro

Técnico Científico - PUC-Rio

Rio de Janeiro, 18 de julho de 2008

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

Fernando Diniz Hämmerli

Engenheiro Eletricista com ênfase em Telecomunicações, graduado em 2004 pela Pontifícia Universidade Católica do Rio de Janeiro. Atual linha de pesquisa: Desenvolvimento de dispositivos para redes de alta capacidade e disponibilidade, baseados em circuitos eletrônicos de lógica programável.

Ficha Catalográfica

Hämmerli, Fernando Diniz

Desenvolvimento de um analisador de elementos de rede baseado no padrão Gigabit Ethernet / Fernando Diniz Hämmerli ; orientador: Marbey Manhães Mosso. – 2008.

81 f. il. (color.) ; 30 cm

Dissertação (Mestrado em Engenharia Elétrica)– Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, 2008.

Inclui bibliografia

1. Engenharia elétrica – Teses. 2. Rede. 3. Elemento de rede. 4. Gigabit Ethernet. 5. Máxima taxa de transmissão. 6. Perda de quadros. 7. Retardo. 8. VHDL. I. Mosso, Marbey Manhães. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. III. Título.

CDD: 621.3

A todos os meus amigos, colegas, familiares e professores que sempre me apoiaram, incentivaram e acreditaram no sucesso deste trabalho.

Aos meus colegas do RDC, Prof. Oliveira, José Telles, Paula Cyrillo, Diogo Assis, Nelson Riet e Christiano Muniz por todo apoio dispensado.

E, em especial, à minha esposa, Sandra, por toda confiança e compreensão nos momentos mais difíceis ao longo desses dois últimos anos.

Agradecimentos

Ao meu orientador, Professor Marbey Manhães Mosso, pelo incentivo e parceria na realização deste trabalho.

Ao pesquisador e colega Henrique Portella da Silva pela grande contribuição no desenvolvimento deste projeto.

Aos meus colegas do GSOM/CETUC por todo apoio técnico dispensado.

Aos professores que participaram da Comissão examinadora.

À Paula Cyrillo Gomes pelo auxílio na revisão deste texto.

Resumo

Hämmerli, Fernando Diniz; Mosso, Marbey Manhães. **Desenvolvimento de um analisador de elementos de rede baseado no padrão Gigabit Ethernet.** Rio de Janeiro, 2008. 81p. Dissertação de Mestrado - Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

O objetivo desta dissertação consiste no desenvolvimento e na realização de um analisador de redes e elementos de redes na taxa de 1 Gbps. A tecnologia de lógica programável (FPGA) é utilizada através de uma placa de desenvolvimento ativada por ferramentas computacionais dedicadas a esta aplicação. O módulo realizado é utilizado para caracterizar uma rede e alguns elementos de rede em função da taxa máxima de transmissão de pacotes, número de pacotes perdidos e retardo. Uma comparação entre os resultados fornecidos pelo módulo desenvolvido e um equipamento comercial é apresentada e comentada. Finalmente, as principais vantagens da proposta desta dissertação são destacadas.

Palavras-chave

Rede; elemento de rede; Gigabit Ethernet; máxima taxa de transmissão; perda de quadros; retardo; VHDL

Abstract

Hämmerli, Fernando Diniz; Mosso, Marbey Manhães. **Development of a Gigabit Ethernet element analyzer**. Rio de Janeiro, 2008. 81p. Master's Thesis – Electrical Engineering Department, Pontifícia Universidade Católica do Rio de Janeiro.

The main purpose of this dissertation is the development and realization of a Gigabit Ethernet network element analyzer. The FPGA technology is employed through a development board, activated by dedicated software tools. The prototype realized is employed to describe a network and network elements by maximum transmission capacity, frame loss and delay. A comparative analysis between this prototype and a commercial equipment is performed. Finally, the main advantages of this dissertation will be highlighted.

Keywords

Network; network element; transmission rate; packet loss; delay; VHDL

Sumário

1	Introdução	14
2	Avaliação de desempenho de uma rede de telecomunicações	17
2.1.	Introdução	17
2.2.	Principais parâmetros de desempenho	18
2.2.1.	Perda de quadros	19
2.2.2.	Atraso ou retardo	20
2.2.3.	Variação do retardo ou <i>jitter</i>	21
2.2.4.	Máxima capacidade de transmissão	22
2.2.5.	Comentários e conclusões	22
3	Tecnologia FPGA	23
3.1.	FPGA: Histórico, linguagens e blocos	23
3.2.	Componentes lógicos de comunicação em <i>chips</i> FPGA	24
3.2.1.	Interface <i>Local-link</i>	25
3.2.2.	Barramento OPB	26
3.3.	Comentários finais e conclusões	27
4	Arquitetura básica de um analisador de elementos de redes	28
4.1.	Introdução	28
4.2.	Avaliação da perda de quadros	29
4.3.	Avaliação da máxima capacidade de transmissão	32
4.4.	Avaliação do retardo sobre os quadros	33
4.5.	Comentários e conclusões	38
5	Realização do equipamento	39
5.1.	Detalhes da tecnologia adotada	39
5.1.1.	O Processador embarcado Power-PC	40
5.1.2.	DCM - <i>Digital Clock Manager</i>	42

5.1.3. MGT - Multi-Gigabit Transceiver	43
5.1.4. PLB – Processor Local Bus	44
5.2. Módulos	45
5.2.1. Módulo de transmissão	45
5.2.1.1. Cálculo do intervalo entre quadros (IFG)	48
5.2.1.2. Capacidade máxima de transmissão	50
5.2.1.3. Fluxograma do módulo de transmissão	51
5.3. Módulo de recepção	53
5.3.1. Fluxograma do módulo de recepção	56
5.3.2. Envio de quadros para aprendizado	58
5.4. Controle e comunicação externa	58
5.4.1. Isolamento do sinal de <i>clock</i> do barramento OPB	60
5.4.2. A gerência através de <i>software</i>	62
5.5. Programa de gerenciamento e interface com o usuário	63
5.6. Comentários e conclusões	66
6 Testes realizados	67
6.1. Descrição do teste	67
6.2. Descrição do elemento de rede escolhido	67
6.3. Teste de referência	69
6.4. Resultados obtidos	69
6.5. Fotos da montagem para a realização dos testes	75
6.6. Comentários e Conclusões	76
7 Comentários finais e conclusões	77
7.1. Observações finais	77
7.2. Principais resultados obtidos	77
7.3. Principais problemas e desvantagens	78
7.4. Comentários adicionais e trabalhos futuros	79
8 Referências	80

Lista de figuras

Figura 1 – Diagrama básico da interface <i>local-link</i>	25
Figura 2 – Diagrama de conexão da interface <i>local-link</i>	26
Figura 3 – Organização estrutural (simplificada) da solução	28
Figura 4 – Inclusão no quadro de dados relativos ao teste	30
Figura 5 – Quadro contendo sinalizador de início de teste	31
Figura 6 – Fluxograma da lógica do receptor, capaz de contabilizar a perda de quadros	32
Figura 7 – Quadro contendo dados que possibilitam o cálculo do retardo	34
Figura 8 – Organização estrutural da solução capaz de avaliar o retardo	34
Figura 9 – Fluxograma da lógica do receptor para calcular o retardo médio dos quadros	36
Figura 10 – Bloco DCM	42
Figura 11 – Diagrama dos barramentos PLB e OPB, e a ponte que permite a comunicação entre eles	45
Figura 12 – Diagrama do módulo transmissor – primeira versão proposta	46
Figura 13 – Diagrama do módulo transmissor – segunda versão proposta, incluindo informações necessárias para o cálculo do retardo.	48
Figura 14 – Relação entre a taxa de transmissão e o IFG, para diferentes tamanhos de quadro	51
Figura 15 – Fluxograma da lógica do módulo transmissor, que descreve a sua máquina de estados	52
Figura 16 – Diagrama do módulo receptor – primeira versão proposta	53
Figura 17 – Diagrama do módulo receptor – segunda versão proposta, capaz de calcular o retardo dos quadros	55
Figura 18 – Diagrama detalhado da lógica do módulo receptor, que descreve a sua máquina de estados	57
Figura 19 – Diagrama do módulo de controle	59
Figura 20 – Diagrama do mecanismo de troca de informações entre pontos alimentados por sinais de <i>clock</i> distintos	61
Figura 21 – Detalhe do funcionamento do semáforo proposto	62

Figura 22 – Tela principal do programa de administração	64
Figura 23 – Tela principal do programa de administração, exibindo resultados parciais, durante um teste	65
Figura 24 – Tela de configuração do acesso pela porta serial	65
Figura 25 – Tela de configuração do período entre atualizações durante a realização de um teste	66
Figura 26 – Diagrama de blocos da montagem, para a realização dos testes	68
Figura 27 – Representação do resultado em forma gráfica, relacionando a máxima capacidade de transmissão, em quadros por segundo, com o tamanho dos quadros	70
Figura 28 – Representação do resultado em forma gráfica, relacionando a máxima capacidade de transmissão, em taxa de bits por segundo, com o tamanho dos quadros	71
Figura 29 – Representação gráfica dos resultados em comparação com os limites teóricos, em taxa de quadros e escala linear	71
Figura 30 – Representação gráfica dos resultados em comparação com os limites teóricos, em taxa de quadros e escala logarítmica	72
Figura 31 – Representação gráfica dos resultados em comparação com os limites teóricos, em taxa de bits e escala linear	72
Figura 32 – Representação gráfica dos resultados em comparação com os limites teóricos, em taxa de bits e escala logarítmica	73
Figura 33 – Representação do resultado em forma gráfica, relacionando o retardo médio, medido por ambos os equipamentos, com o tamanho dos quadros	74
Figura 34 – Vista geral da montagem durante a realização dos testes	75
Figura 35 – Detalhe da placa principal e do módulo óptico externo	75

Lista de tabelas

Tabela 1 – Modelo de matriz de resultados obtidos pelo módulo receptor, relacionando o retardo médio em faixas de tempo do segundo anterior.	37
Tabela 2 – Velocidades de operação de alguns padrões de comunicação, que podem ser obtidas com o uso do MGT	43
Tabela 3 – Máxima capacidade de transmissão, no padrão Gigabit-Ethernet, para quadros de diferentes tamanhos	50
Tabela 4 – Resultado dos testes, representando máxima capacidade de transmissão em quadros por segundo	70
Tabela 5 – Resultado dos testes, representando o retardo médio medido	74

Abreviaturas e siglas

ARP	Address Resolution Protocol
ATM	Asynchronous Transfer Mode
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
DCM	Digital Clock Manager
FPGA	Field Programmable Gate Array
HDL	Hardware Description Language
IEEE	Institute of Electrical and Electronics Engineers
IFG	Inter-Frame Gap
IPTV	Internet Protocol Television
MAC	Media Access Control
MGT	Multi-Gigabit Transceiver
OPB	On-chip peripheral bus
PDH	Plesiochronous Digital Hierarchy
PLB	Processor Local Bus
QoS	Quality of Service
RAM	Random access memory
RFC	Request for Comments
ROM	Read Only Memory
SDH	Synchronous Digital Hierarchy
SFP	Small Form-Factor Pluggable
SLA	Service Level Agreement
SONET	Synchronous Optical Network
UART	Universal asynchronous receiver/transmitter
VHDL	VHSIC hardware description language
VHSIC	Very High Speed Integrated Circuit
VOIP	Voice Over IP
Wi-Fi	Wireless Fidelity

1 Introdução

Em maio de 1973, Rob Metcalfe, um pesquisador da Xerox Palo Alto Research Center (PARC), na Califórnia, Estados Unidos, divulgou um memorando onde descrevia um sistema de rede para interconexão de estações de trabalho, com ênfase para a capacidade de transmissão de dados para impressoras de alta velocidade. No PARC também estavam em desenvolvimento, impressoras para computadores pessoais, um microcomputador com interface gráfica e um dispositivo de entrada, então inédito, chamado de mouse. A intenção do projeto era criar um ambiente de alta capacidade computacional, de baixo custo e acessível a qualquer empresa. O padrão de rede, chamado Ethernet era uma evolução da rede Aloha[1], criada pela Universidade do Havaí, no final da década de 60. A rede Aloha era voltada para comunicação através de rádio. Metcalfe melhorou a técnica original do Aloha de arbitragem de acesso ao canal e desenvolveu um mecanismo de detecção de colisão, para uso em meio confinado. O nome Ethernet foi derivado da palavra *ether*, termo usado no passado para descrever um suposto meio necessário para a propagação de ondas eletromagnéticas. O nome escolhido também objetivava deixar claro que se tratava de um padrão que poderia ser usado por qualquer tipo de computador, e não apenas os desenvolvidos no PARC.

O padrão Ethernet original foi publicado em 1980, pelo consórcio de empresas DEC-Intel-Xerox, e estabelecia uma taxa de transmissão de 10 Mbps. Em 1985 o IEEE assumiu o padrão original como base para o padrão intitulado “IEEE 802.3 Carrier Sense Multiple Access with Collision Detection (CSMA/CD)” [2]. O principal atrativo do padrão Ethernet, e certamente um dos motivos do seu sucesso, foi a sua simplicidade aliada ao baixo custo. Em meados da década de 80, computadores pessoais já eram uma realidade, principalmente nos países mais desenvolvidos. Uma avalanche de esforços voltados para mercados corporativos e domésticos colocavam à disposição dos consumidores inúmeras aplicações

atrativas. Novas ferramentas de desenvolvimento, aliadas ao crescente número de profissionais qualificados no mercado, permitiram às empresas de médio e até pequeno porte, criar aplicativos específicos e personalizados. Nesse ambiente, o padrão Ethernet encontrou um meio absolutamente propício para expansão.

A utilização do padrão Ethernet no modo *Full-Duplex*, tornou esta tecnologia ainda mais simples e barata. Nos anos que se seguiram, as aplicações dedicadas às redes locais migraram para as redes metropolitanas e de longa distância. Iniciou-se então, no fim da década de 90, um cenário competitivo com as tecnologias TDM utilizadas pelas grandes operadoras telefônicas.

As redes telefônicas e as redes de computadores sempre seguiram caminhos distintos. As redes digitais de telefonia evoluíram para os padrões PDH e SDH/SONET[1], específicos para transmissão de voz e com alta qualidade. Em meados dos anos 90 foi criado o padrão ATM[1], que tinha como objetivo atender aos dois tipos de tráfego, preservando a garantia de qualidade. Ele teve uma adoção relativamente alta, porém, assim como o SDH/SONET, também é complexo e possui elevado custo, tanto de implantação quanto de operação. Tal custo é consequência direta das rígidas exigências de qualidade de serviço (QoS), utilizados pelas operadoras telefônicas.

Com o surgimento de aplicações de voz sobre IP (VoIP)[1] oferecido por provedores de internet, o mercado de serviços na área de telecomunicações começou a apresentar mudanças. A utilização de um SLA (*Service Level Agreement*) com padrões de qualidade de serviço (QoS) flexíveis, permitiu que tecnologias mais simples, especialmente o Ethernet, se tornassem cada vez mais presentes em redes metropolitanas. Diante desse cenário, diversos fabricantes de equipamentos apresentaram módulos dedicados à avaliação de desempenho destas novas redes. Entretanto, a maioria destes módulos baseia seus testes em camadas superiores, envolvendo aplicações TCP/IP, tornando esses equipamentos de medição mais complexos e com custos mais elevados.

Considerando-se então o cenário apresentado, a presente dissertação tem como principais objetivos os seguintes tópicos:

- Desenvolvimento de um módulo de testes capaz de avaliar o desempenho de conexões e elementos de redes Ethernet na taxa de 1 Gbps;
- Caracterizar redes e elementos de rede, com testes baseados nas camadas inferiores, permitindo assim o desenvolvimento de analisador simples e de baixo custo;
- Utilizar a tecnologia FPGA no presente desenvolvimento para permitir adaptações futuras destas aplicações às exigências do mercado.

A organização do presente texto contará com 7 capítulos. O capítulo 1 refere-se à presente introdução. Ao longo do capítulo 2 são introduzidos conceitos básicos dedicados a avaliação do desempenho de conexões e redes de telecomunicações. Ao longo do capítulo 3 é introduzida a tecnologia FPGA utilizada no desenvolvimento desta dissertação. Ao longo do capítulo 4, um analisador de redes e de elementos de redes é proposto para desenvolvimento, destacando sua arquitetura e funcionalidades. Ao longo do capítulo 5 é apresentada uma descrição detalhada da realização do módulo de teste, destacando os detalhes da tecnologia adotada. Ao longo do capítulo 6 são apresentados os testes realizados e os resultados obtidos. Finalmente, no capítulo 7 são apresentados os principais comentários e conclusões da presente dissertação.

2

Avaliação de desempenho de uma rede de telecomunicações

Ao longo do presente capítulo são introduzidos os principais elementos qualitativos e quantitativos capazes de permitir a avaliação do desempenho de uma rede de telecomunicações.

2.1. Introdução

Sempre que uma rede de telecomunicações é projetada, uma das principais dificuldades encontradas é o atendimento das características de desempenho desejadas. Uma vez que o projeto é executado, pode ser essencial a realização de testes probatórios de qualidade. Muitas vezes, os engenheiros precisam ensaiar determinados cenários de operação, mais críticos, e observar se os resultados obtidos atendem às especificações desejadas ou são compatíveis com simulações realizadas. Estes testes podem ser utilizados para avaliar desde grandes trechos de rede até dispositivos individuais.

Basicamente os testes de desempenho são realizados com a injeção de um determinado tráfego na rede e a observação da resposta da rede a este tráfego. O mecanismo de geração deste tráfego pode ser simples, ou obedecer a padrões complexos. Vários parâmetros de desempenho podem ser aferidos, como a taxa de transmissão máxima suportada, o tempo que um quadro levou em trânsito, ou quanto tempo a rede precisa para se recuperar de uma falha. Geralmente, o módulo responsável por esta análise deve conhecer as características do tráfego que foi gerado e, assim, ser capaz de aumentar a precisão dos seus resultados.

Redes com características de tráfego peculiares também são candidatas a testes rigorosos, como por exemplo, em casos de redes com elevado tráfego de voz misturado a tráfego de dados. Como o tráfego de voz é sensível às variações

de desempenho[3], o operador da rede deve garantir que este tráfego não será afetado mesmo sob as condições mais severas. Nos próximos anos as redes de IPTV estarão presentes em quase todas as cidades e, há algum tempo, grandes empresas têm conduzido estudos de qualidade para estas redes. Assim como o tráfego de voz, o tráfego de vídeo também requer uma qualidade de transmissão elevada, além de exigir uma taxa muito superior. O problema do tráfego de vídeo se agrava quando utilizado de forma interativa, como em videoconferências, aulas à distância, etc.

O protocolo Ethernet[2] tem sido utilizado e sua penetração nas aplicações de redes de telecomunicações está se intensificando. Em muitas aplicações, redes virtuais ou via túneis fornecem a um cliente residencial ou corporativo uma extremidade de um cabo. A forma como o serviço é estruturado dá ao usuário final a sensação de possuir uma rede única e privada. A operadora se comprometerá com seu cliente, através de um acordo de qualidade, ou SLA (*Service Level Agreement*), a atender requisitos mínimos de desempenho na prestação daquele serviço. O cliente poderá, caso julgue necessário, testar se aqueles parâmetros de qualidade estão sendo respeitados.

2.2. Principais parâmetros de desempenho

Existem inúmeros parâmetros, testes e dados referenciais para avaliação de desempenho dos equipamentos de redes. A escolha da melhor configuração de teste pode variar drasticamente conforme o ambiente envolvido ou resultados esperados. Testes altamente críticos em um cenário podem ser de menor importância em outro, cabendo aos responsáveis pela avaliação determinar, não apenas o ferramental técnico usado, mas também a metodologia que irá conduzir o trabalho.

Alguns parâmetros, no entanto, são consensualmente valorizados e, devido a este elevado reconhecimento, são amplamente utilizados. Neste tópico, vamos citar rapidamente alguns deles, pois terão grande importância no desenvolvimento da presente dissertação. É importante citar que, para a grande maioria dos

parâmetros de desempenho, as causas responsáveis pela sua degradação podem estar concentradas em um único ponto ou espalhadas pelos diversos elementos responsáveis pela transmissão de dados. Um trabalho que objetive unicamente a aferição do desempenho pode não exigir a identificação dos pontos responsáveis pela sua degradação. Outros testes poderão exigir como produto, não apenas a identificação destes pontos, mas também minuciosas descrições das condições envolvidas e até mecanismos de resolução ou mitigação dos efeitos indesejados.

A maioria dos testes costuma ser realizada iterativamente com variação discreta e conhecida do padrão de transmissão. São comumente encontrados testes que relacionam os resultados com a taxa de transmissão empregada e o tamanho dos quadros. Também podem ser usados padrões pseudo-aleatórios, controlados, para simular ambientes específicos. Por exemplo, podem ser gerados quadros com características semelhantes aos de uma rede de voz, permitindo avaliar os efeitos obtidos sobre o tráfego. Podem então ser destacados os seguintes parâmetros, abaixo relacionados.

2.2.1. Perda de quadros

Do inglês, *Frame Loss*, refere-se à taxa de quadros perdidos entre dois pontos da rede, e costuma ser representado em notação porcentual ou fracionada. O mecanismo de detecção de perda de quadros pode usar lógicas distintas, conforme o cenário. Avaliar se um quadro foi perdido, depende do entendimento de como esse quadro poderia ser recebido. Imagine um teste em que quadros são transmitidos com números seqüenciais de controle. Em uma rede onde não ocorre reordenação de quadros, o recebimento de um quadro com numeração superior ao imediatamente posterior ao último recebido, indica a falha. Em uma rede onde a reordenação é possível, outros mecanismos de avaliação seriam necessários. O mais simples envolve o uso de um contador de quadros. Ao fim do teste a diferença entre o número de quadros transmitidos e recebidos indica o número de quadros perdidos. Quadros também podem ser considerados como perdidos se não forem recebidos após um determinado intervalo de tempo da sua transmissão.

Um ponto importante a mencionar é o fato de que quadros podem ser perdidos, por exemplo, por descarte quando a fila de um equipamento está cheia ou por falha na verificação do código de detecção de erro. No segundo caso, o quadro pode ter sido entregue ao seu destino final, mas será igualmente contabilizado na estatística de perda.

2.2.2. Atraso ou retardo

Outro importante parâmetro de desempenho, conhecido como atraso, refere-se ao tempo decorrido desde a transmissão de um quadro até seu efetivo recebimento no destino. O atraso total é, na verdade, a soma de uma série de atrasos, consequência de todos os processos envolvidos na transmissão daquela informação. Alguns desses componentes têm maior significância do que outros sobre o resultado final.

Como exemplos, podemos citar a taxa de transmissão de bits empregada nos enlaces ao longo do trajeto, o tempo gasto pelos equipamentos ativos para processar o quadro e o tempo que o quadro permaneceu armazenado em filas de recepção e transmissão desses equipamentos. Outro fator que deve ser considerado é o tempo de propagação do sinal transmitido. Em redes de curta distância e com taxas de transmissão reduzidas, este parâmetro assume dimensões diminutas em relação aos demais e, por isso, costuma ser desprezado. Em redes de longa distância e com taxas de transmissão muito elevadas, o tempo gasto pelo sinal para percorrer um meio de transmissão se torna comparável, em ordem de grandeza, aos demais efeitos causadores de atraso e, portanto, deve ser considerado.

Considerando-se os usuários e os aplicativos, o atraso pode exercer grande impacto negativo sobre o resultado obtido se estiver acima dos padrões exigidos. As aplicações mais sensíveis ao atraso são aquelas que trabalham de forma interativa, por exemplo, voz, videoconferência ou terminais remotos.

2.2.3. Variação do retardo ou *jitter*

Supondo que o tráfego e todos os componentes, ativos e passivos, da rede trabalhem constantemente sob um regime estável, é de se esperar que o atraso obtido sobre os quadros seja constante. Na prática, toda a transmissão está sujeita a inúmeras variações e, entre elas, a variação no comprimento das filas dos equipamentos. Como consequência, obtém-se uma seqüência de quadros recebidos com variação no atraso entre eles, efeito conhecido, em inglês, como *jitter*. O efeito também poderá ser causado em redes onde ocorram multi-percursos, ou seja, quadros seguindo caminhos distintos e, conseqüentemente, com retardos diferentes.

A variação do atraso pode ser extremamente danosa aos serviços interativos. O motivo está no fato que, quanto maior for a variação, mais imprevisível será o momento do recebimento do próximo quadro. Quando a variação é nula, ou seja, o atraso é constante, os quadros são recebidos com o mesmo intervalo com que foram transmitidos. O *jitter* implica no recebimento dos quadros com intervalos distintos aos da sua transmissão.

Aplicativos de voz ou videoconferência, quando transportados através de uma rede de dados, utilizam filas[4] de armazenamento temporário como forma de mitigar os efeitos da variação de retardo. A fila protege o serviço da variação em duas situações: armazenar quadros recebidos antes do seu instante de processamento e garantir uma reserva de quadros a processar para suprir períodos em que eles sejam recebidos com taxa inferior à necessária. O dimensionamento das filas é importante para aperfeiçoar o desempenho da aplicação. Se a fila for curta demais ela poderá não servir a seu propósito, porém se for grande demais poderá criar um retardo inadequado ao serviço.

2.2.4. Máxima capacidade de transmissão

Na análise de desempenho de uma rede, um dos principais objetivos é o conhecimento da capacidade máxima de transmissão[5] associada. É a partir dessa informação que os técnicos e engenheiros responsáveis poderão afirmar se o objeto do teste, seja um único equipamento ou um segmento de rede, será capaz de suportar um determinado tráfego desejado. Grande parte dos fabricantes de equipamentos de rede, sejam comutadores (*switches*) ou roteadores, costuma divulgar em seus manuais as especificações de tráfego suportadas. Estas informações são de vital importância, mas não oferecem precisão para estimativas envolvendo arranjos de vários equipamentos numa rede, nem o comportamento real daquele equipamento quando submetido a tráfegos com características peculiares.

O teste de máxima capacidade objetiva descobrir qual a maior taxa que pode ser transmitida através da rede ou de um equipamento, sem a interferência dos efeitos negativos já descritos, especialmente a perda de pacotes. Os métodos usados, em sua grande maioria, realizam sucessivos testes por curtos espaços de tempo, variando de forma discreta as taxas de transmissão. O teste completo costuma ser repetido inúmeras vezes, sendo aplicadas variações em outros parâmetros como, por exemplo, o tamanho do quadro transmitido.

2.2.5. Comentários e conclusões

Ao longo do presente capítulo foram introduzidas e comentadas as principais características das redes de telecomunicações associadas ao seu desempenho. Além de considerações qualitativas, parâmetros descrevendo a perda de quadros, retardo, *jitter* e a máxima capacidade de transmissão foram também introduzidos para permitir que o desempenho quantitativo fosse também avaliado.

3 Tecnologia FPGA

Ao longo do presente capítulo será apresentada uma descrição introdutória da tecnologia FPGA e dos módulos básicos que a constitui.

3.1. FPGA: Histórico, linguagens e blocos

Muitos dos circuitos integrados em nosso dia-a-dia vêm programados de fábrica, destinados a aplicações específicas, ou seja, sua programação é realizada no ato da fabricação, fisicamente, sem a possibilidade de alteração posterior. O custo fixo de produção deste tipo de dispositivo é extremamente alto, ou seja, o processo para a criação da primeira unidade é milhares, ou milhões, de vezes mais caro que a produção individual das demais. Portanto, só é justificado produzi-lo se for utilizado para produção de dispositivos em massa. Outra desvantagem de seu uso é que toda a sua programação é definida durante seu projeto, não sendo possível qualquer alteração depois que a matriz é criada. Caso seja encontrado um erro no projeto, será necessário descartar toda a produção.

Em 1985 uma empresa Americana, a Xilinx Inc, apresentou um novo modelo de *chip*, capaz de ser reprogramado de acordo com as aplicações do usuário, chamado de FPGA[6], abreviação de “*Field Programmable Gate Array*”. O FPGA é composto, basicamente, por:

- blocos lógicos: circuitos idênticos, constituídos pelo agrupamento de *flip-flops*, permitem a criação de elementos lógicos funcionais;
- blocos de entrada e saída: permitem o interfaceamento dos sinais entre os blocos lógicos e a saída do circuito;
- matriz de interconexão: são chaves e trilhas usadas para interligar os blocos. Este processo, conhecido como roteamento, é bastante complexo e definido pelo software usado na programação do *chip*. Se for mal realizado, a lógica pode não funcionar corretamente;

Um *chip* FPGA contém uma enorme quantidade de blocos lógicos. Através do roteamento entre eles, é possível a implantação de lógicas bastante complexas. O desenvolvimento de módulos, ou dispositivos, nos *chips* FPGA geralmente é feito através de linguagens de descrição de hardware, do inglês “*Hardware Description Language*” ou, simplesmente, HDL. À primeira vista semelhantes a linguagens de computador tradicionais, as linguagens HDL são na verdade um conjunto de instruções que permitem aos desenvolvedores descrever a lógica desejada para o hardware de forma mais intuitiva e rápida. Uma vez concluído, o código passa por um processo de compilação, ou tradução, sendo então implantando no *chip*. As linguagens mais conhecidas são VHDL[7] e Verilog[8].

Existem inúmeras aplicações para os *chips* FPGA, sendo que algumas são tão comuns que os fabricantes desenvolvem modelos especializados em determinadas funções, como para redes de comunicação ou processamento de sinais. Vários modelos de *chips* FPGA, como o usado neste trabalho, possuem microprocessadores embutidos de forma que é possível executar programas em paralelo com a lógica do FPGA. É curioso ainda salientar que, mesmo em *chips* FPGA de menor custo, onde não há a presença de processadores, é possível implantar um através de programação de lógica física. Vários fabricantes possuem versões já prontas de seus processadores, que podem ser implantadas se necessário. Obviamente, processadores “criados” com lógica no FPGA jamais possuirão eficiência comparável a de processadores fabricados diretamente no *chip*. É, portanto, importante frisar que existe um nicho de mercado ideal para os circuitos FPGA, onde existe demanda por desempenho, mas não é viável o desenvolvimento de *chips* dedicados.

3.2.

Componentes lógicos de comunicação em *chips* FPGA

Quaisquer projetos desenvolvidos em circuitos FPGA, mesmo com lógicas diminutas, devem ser estruturados de forma modular. Esta abordagem contribui com a simplicidade do projeto, além de permitir reuso destes blocos inúmeras vezes dentro do mesmo circuito. A comunicação entre esses blocos lógicos pode

ser realizada diretamente, através da interconexão direta dos mesmos, ou por mecanismos mais inteligentes, com controles de transmissão ou através de barramentos de dados. Abaixo serão descritos, de forma sucinta, dois mecanismos de comunicação de grande importância na realização deste trabalho:

3.2.1. Interface *Local-link*

Este padrão foi desenvolvido visando a criação de canais de comunicação entre módulos, idealmente em dispositivos de redes. Admite o uso de largura de dados variável, permitindo ajuste à taxa de transmissão desejada, tamanho variável dos quadros, comunicação bidirecional e síncrona, além de possuir controle de fluxo.

A figura abaixo mostra o diagrama básico de funcionamento da interface *local-link*[9].

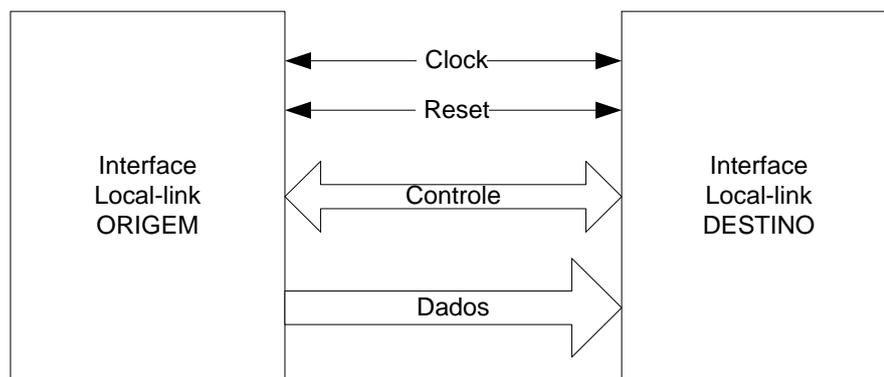


Figura 1 - Diagrama básico da interface *local-link*

O uso comum dos sinais de *clock* e *reset* é uma consequência do modo de operação síncrona. A interface *local-link* pode ser usada entre quaisquer módulos que necessitem transferir o conteúdo de quadros entre si. No projeto, objeto deste trabalho, a *local-link* será usada entre os módulos de transmissão/recepção de quadros e os módulos responsáveis pela camada de enlace do padrão Ethernet.

A figura abaixo ilustra a organização de um dispositivo com várias portas Ethernet. Para cada uma delas, é usada uma interface *local-link* para realizar a comunicação com o módulo central de comutação de quadros ou pacotes.

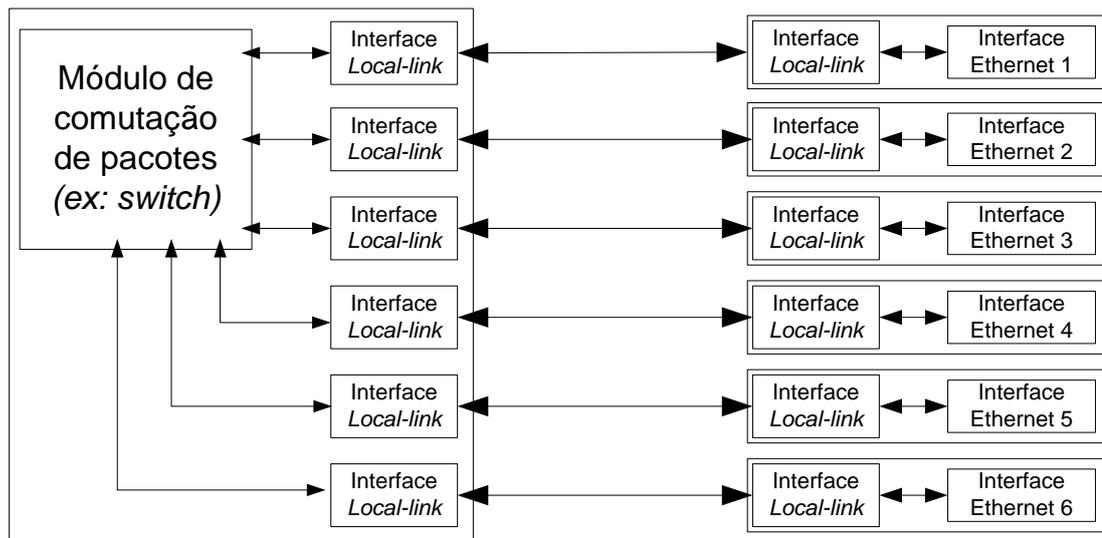


Figura 2 - Diagrama de conexão da interface *local-link*

3.2.2. Barramento OPB

Ao contrário da interface *local-link*, onde as transferências são realizadas ponto-a-ponto e com elevada taxa de transmissão, o OPB[10], sigla do inglês “*On-chip peripheral bus*”, oferece um canal de comunicação em barra conectado a diversos módulos, ou dispositivos, implantados no mesmo *chip*. Dispositivos conectados ao barramento OPB podem transferir dados entre si usando este canal comum. Desenvolvido pela IBM, o OPB tem como ponto forte o seu mecanismo de organização, responsável por evitar problemas em seu funcionamento e, especialmente, controlar as permissões e privilégios de acesso ao meio. Esse controle é realizado por um dispositivo ativo, chamado de árbitro.

Por ter sido desenvolvido para trabalhar com taxas mais reduzidas de transmissão, tipicamente bem inferiores às atingíveis com o *local-link*, o OPB não se mostra adequado para a transferência de quadros, porém é bastante atraente para operações de controle e gerência, devido à sua simplicidade e capacidade de interconectar diversos dispositivos. O OPB também pode permitir a comunicação com outros dispositivos de comunicação de baixa velocidade, como portas seriais, paralelas, modems analógicos, telas, ou até mesmo módulos de memória RAM, ROM ou flash.

3.3. Comentários finais e conclusões

Ao longo do presente capítulo foram introduzidos os tópicos básicos da tecnologia FPGA, destacando-se as linguagens utilizadas (VHDL e Verilog), os blocos lógicos, os blocos de entrada e saída, a matriz de interconexão e as interfaces *local-link* e OPB. Alguns comentários envolvendo a programação e utilização destes blocos em redes de comunicação e no gerenciamento e controle das funcionalidades envolvidas foram adicionados.

4

Arquitetura básica de um analisador de elementos de redes

Neste capítulo é apresentado o desenvolvimento de um dispositivo analisador de redes e de elementos de redes, utilizando tecnologia FPGA. Conforme foi mencionado no capítulo anterior, a realização de soluções nessa tecnologia é bastante simplificada com a divisão das tarefas em módulos ou blocos, e com a comunicação entre os mesmos.

4.1. Introdução

O aparato proposto terá como função básica a geração de um determinado fluxo conhecido de quadros, que é injetado em uma rede ou um equipamento e posteriormente recebido. A análise do fluxo recebido, em comparação com o fluxo transmitido, possibilita a avaliação dos parâmetros de desempenho. O processamento dessas informações e a posterior contabilidade dos resultados, é realizada por um terceiro módulo de controle, também implantado no *chip* FPGA.

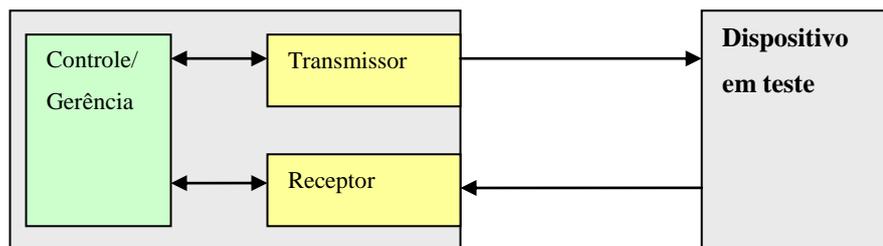


Figura 3 – Organização estrutural (simplificada) da solução

A figura acima ilustra, de forma bastante resumida, a organização estrutural proposta, onde os módulos “Transmissor” e “Receptor” são supervisionados pelo módulo “Controle e Gerência”. Este, por sua vez, através da realimentação fornecida pelos dois outros módulos é capaz de verificar os resultados obtidos durante o teste.

O mecanismo adotado é apresentado em etapas, numa abordagem evolutiva, focada nos testes de desempenho, no intuito de ampliar a compreensão do processo.

4.2. Avaliação da perda de quadros

A avaliação da perda de quadros pode ser pensada de forma extremamente simplificada. Em poucas palavras, se dá através da transmissão de uma determinada quantidade de quadros, seguida de comparação com o número de quadros recebidos. O maior problema de uma solução deste tipo é garantir o perfeito sincronismo entre transmissor e receptor. Este precisa estar ciente do início de um teste, quando seu contador de quadros recebidos deve estar zerado. Ainda assim, a mera contagem de quadros recebidos pode ser problemática se este não for capaz de distinguir um quadro efetivamente enviado pelo módulo transmissor de um quadro com outra origem, que lhe foi entregue. Outra situação potencialmente perigosa ocorre se, por algum motivo, está se realizando testes em uma rede ou equipamento que possibilite a entrega duplicada de um mesmo quadro. Além disso, durante a realização do teste, não seria possível distinguir, da diferença entre quadros transmitidos e recebidos, quais ainda estão em trânsito e quais efetivamente foram perdidos.

A complexidade aumenta em situações nas quais os módulos de transmissão e recepção não estejam fisicamente conectados num mesmo equipamento. Nestes casos, toda a negociação do teste deve ser feita de maneira externa, ou de forma manual ou usando a própria rede. Entenda-se por negociação, o mútuo conhecimento de parâmetros vitais para a realização do teste, por exemplo, a quantidade de quadros a serem transmitidos. Uma solução que reduz bastante os problemas citados, e de baixa complexidade de realização, constitui-se da inclusão do número do quadro em seu conteúdo. Em outras palavras, no campo de dados de um quadro, alguns bits são usados para representar a numeração daquele quadro. Cada quadro possui uma numeração única, sendo transmitidos sequencialmente.

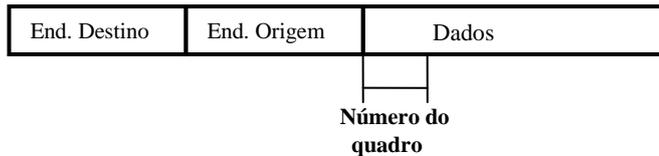


Figura 4 – Inclusão no quadro de dados relativos ao teste

Acima uma representação genérica de um quadro, onde os primeiros bits do campo de dados do mesmo são usados para armazenar a sua numeração. A área de dados é completada com informações não prioritárias, até o tamanho desejado do quadro. Um último detalhe a ser resolvido diz respeito ao conhecimento, por parte do receptor, do início ou fim de um ciclo de teste.

Se os quadros forem transmitidos com numeração seqüencial em ordem decrescente, sendo que o quadro de número um sempre representa o fim do teste, o número do primeiro quadro recebido corresponderá à quantidade de quadros que serão transmitidos naquele teste. Sendo assim, o receptor terá sempre conhecimento da quantidade de quadros restantes para o fim do teste. Mas ainda falta garantir que o receptor tenha pleno conhecimento do início de um novo ciclo de testes. Uma solução é usar um determinado bit na área de dados do quadro para sinalizar que o mesmo corresponde ao primeiro de uma seqüência de teste. Sendo assim, a principal fragilidade do processo descrito se dará quando o primeiro quadro for perdido, pois o receptor não será capaz de identificar o início do teste e, conseqüentemente, irá ignorar os demais quadros. Essa fragilidade pode até mesmo ser desprezada se dois fatores forem considerados.

- Se o primeiro quadro de uma seqüência de testes for perdido, a conseqüência é que o receptor não inicia também o processo de teste, sendo necessária uma intervenção para reiniciar os testes. E, o mais importante, não há resultados comprometidos ou imprecisos.
- Em uma rede com baixa taxa de erro, por exemplo, através de fibras ópticas, a probabilidade do primeiro quadro da seqüência ser perdido é relativamente pequena, o que torna a possibilidade anterior num transtorno incomum.

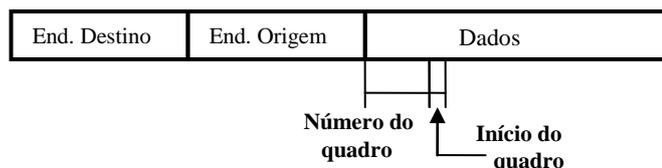


Figura 5 – Quadro contendo sinalizador de início de teste

É importante ressaltar que a negociação descrita até agora ocorre em apenas um sentido de comunicação, do transmissor para o receptor, sendo perfeitamente possível o desenvolvimento de um protocolo de negociação extremamente robusto, com o uso de comunicação bidirecional, que permitiria ao receptor confirmar ao transmissor a ciência dos parâmetros do teste a se realizar. Como o desenvolvimento de um procedimento como este, de maior complexidade, não é objetivo principal deste trabalho, este foi realizado seguindo a proposta anterior.

Nesse cenário, em que todos os quadros recebidos possuem uma identificação, com sua numeração, cabe ao receptor apenas comparar o número do quadro recebido com o número recebido anteriormente. Pela diferença entre os dois números o receptor descobre se algum quadro foi perdido. No caso deste projeto, em que transmissor e receptor se comunicam com um módulo único de controle, no mesmo equipamento, torna-se possível separar, do total de quadros não recebidos, a quantidade efetivamente perdida daqueles ainda em trânsito.

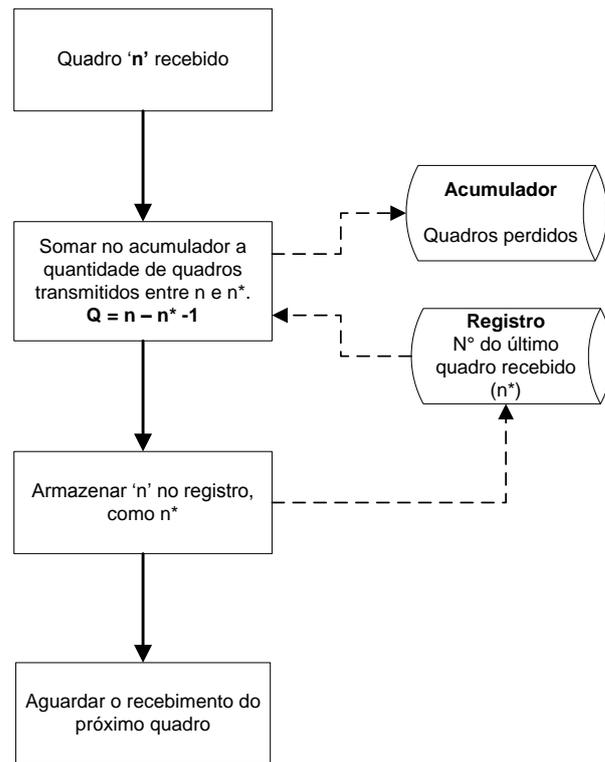


Figura 6 – Fluxograma da lógica do receptor, capaz de contabilizar a perda de quadros

Outro mecanismo importante é o cálculo do número de quadros ainda em trânsito através da numeração proposta, que permite, a partir da taxa de quadros transmitidos, uma estimativa aproximada do retardo introduzido pela rede em teste. Será descrito mais adiante o mecanismo de avaliação precisa deste parâmetro de desempenho.

4.3. Avaliação da máxima capacidade de transmissão

Entenda-se por “máxima capacidade de transmissão” a maior taxa que a rede é capaz de suportar, sem que o fluxo sofra prejuízos além de um determinado limiar definido pelo teste. Um ponto de operação muito procurado é aquele em que o fluxo de quadros não sofre qualquer perda. A lógica mais comum é aquela em que um número grande de quadros é transmitido, sob uma determinada taxa. Se ocorrer perda, significa que o fluxo foi superior à capacidade da rede e o teste deverá ser repetido com taxa menor. Se não ocorreu perda, existe a chance de que

a rede seja capaz de suportar uma taxa ainda maior, e o teste será repetido com taxa superior, até que se encontre o máximo possível.

É importante lembrar que o parâmetro de desempenho que servirá de base para este teste pode ser ajustado conforme a necessidade do usuário. Por exemplo, a tolerância a falhas pode ser maior, aceitando-se fluxos, com até uma determinada quantidade de quadros perdidos, como válidos. Outros parâmetros também podem ser usados, conforme a necessidade do teste. Por exemplo, pode-se avaliar a máxima taxa de transmissão observando-se também o retardo dos quadros. Em redes, ou equipamentos, em que a taxa de transmissão tem grande efeito sobre o retardo, esse segundo parâmetro também pode ser usado como referência para o teste. Não custa lembrar que o retardo tem especial importância sobre determinadas aplicações, especialmente as interativas.

4.4. Avaliação do retardo sobre os quadros

No item 4.1 foi mencionada a possibilidade de se obter uma estimativa aproximada do retardo, com base no número de quadros em trânsito. Esta abordagem pode fornecer uma idéia da ordem do valor real, porém é inadequada se houver necessidade de maior precisão. O principal problema está na impossibilidade de estimativa das pequenas variações de retardo que os quadros experimentam ao longo do trajeto. Desta forma, o cálculo do tempo total utilizando o tempo conhecido, e fixo, entre a transmissão de dois quadros consecutivos, fornecerá um resultado impreciso.

Uma solução viável capaz de atender a esta demanda pode ser feita com o uso de um relógio mestre, comum e único ao transmissor e receptor. Uma solução aparentemente simples, na qual o transmissor insere no conteúdo do quadro, após os campos de numeração e início já mencionados, a posição do relógio no instante da geração daquele quadro. O relógio é, na prática, um contador suficientemente grande para não repetir valores numa mesma rodada de testes ou, pelo menos, garantir um espaço de vários quadros antes de recomeçar sua contagem. Ao

receber o quadro, a diferença entre o instante atual e o marcado no conteúdo do quadro, indicará o retardo sofrido.

A frequência de oscilação deste contador também é de grande relevância. Ela deve ser suficientemente elevada para permitir uma boa precisão, mas se for em demasia exigirá um contador com maior comprimento de bits, aumentando os riscos e dificuldades para implantação no FPGA, devido às exigências mais rígidas de tempo. A precisão utilizada será de $0,8 \mu\text{s}$, correspondendo ao período de um sinal de 125 MHz, disponível no circuito.

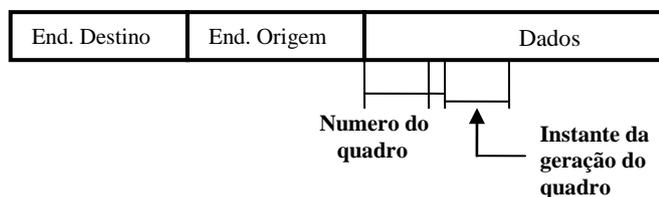


Figura 7 – Quadro contendo dados que possibilitam o cálculo do retardo

Na figura acima, temos a representação do quadro incluindo o campo que contém o valor do relógio no instante em que o quadro é gerado.

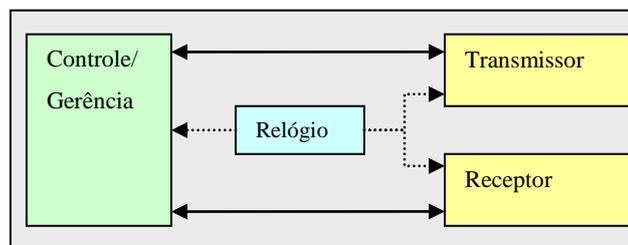


Figura 8 – Organização estrutural da solução capaz de avaliar o retardo

Essa solução nos permite o cálculo preciso do retardo experimentado por um determinado quadro, mas não resolve o nosso problema. Na prática, o que desejamos é de resultados estatísticos para representar o teste realizado. O resultado mais comum é o tempo médio, porém, de posse dos retardos medidos é possível a obtenção de outros resultados. O principal entrave que é enfrentado de imediato é a grande dificuldade de armazenamento de todos os retardos medidos ao longo do teste para posterior processamento. O volume de dados obtidos ao longo de uma rodada de testes é excessivamente grande, e exigiria uma

capacidade de armazenamento externa, em memória RAM, muito elevada. A transmissão para um dispositivo externo, como um computador, é possível, mas exigiria uma interface de comunicação igualmente veloz para ocorrer em tempo real.

Será apresentado um modelo simplificado capaz de estimar a média dos retardos obtidos com base em uma ou mais bases de tempo fixas. Para simplificar, vamos pensar em um bloco capaz de calcular a média dos retardos no último segundo e, a cada segundo, esse valor é transmitido para um computador e o cálculo da média se reinicia. Imediatamente percebemos dois fatos interessantes. Primeiro que a transmissão para o computador requer um desprezível volume de dados, permitindo o uso de interfaces de baixa velocidade, como a porta serial RS-232, de uso bastante simples. Segundo, que ao computador será permitido o armazenamento dessas médias, e posterior análise dos resultados no tempo. O tempo de acúmulo do teste poderá ser reduzido conforme a necessidade do teste e as capacidades técnicas do equipamento utilizado.

Uma forma de realizar esta proposta é com o uso de um acumulador de retardo total e um contador de quadros considerados nesse acumulador. Uma vez que um determinado limite de tempo é atingido, o retardo acumulado é dividido pelo número de quadros e o resultado armazenado em um registro. Este registro, que representa o retardo médio dos quadros no segundo anterior, será lido e transmitido ao computador.

O fluxograma a seguir exemplifica a lógica descrita:

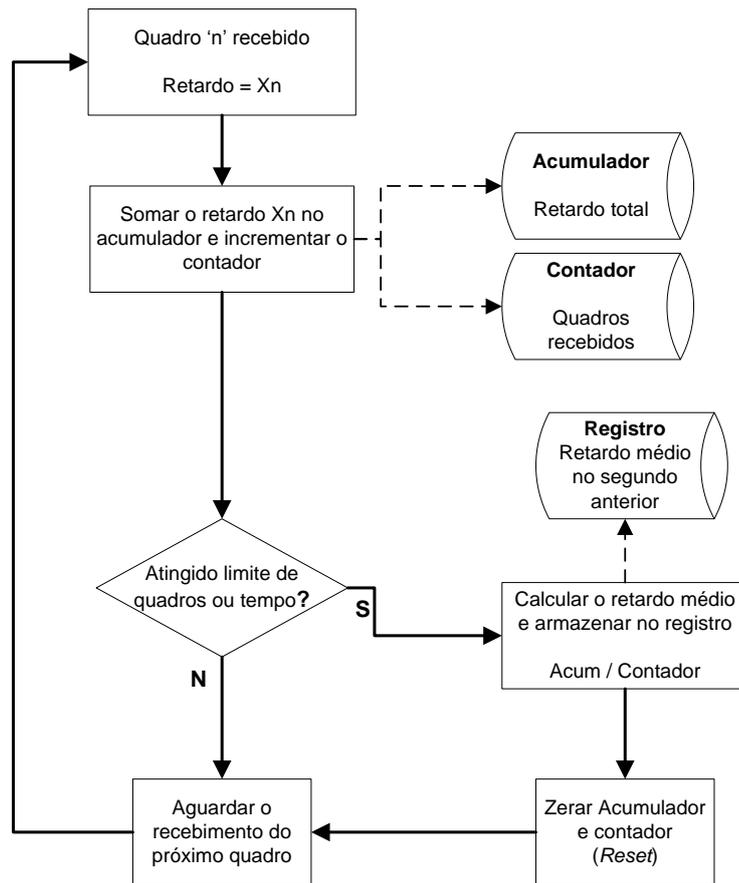


Figura 9 - Fluxograma da lógica do receptor para calcular o retardo médio dos quadros

Por simplicidade, não foi demonstrado o processo de leitura do registro com o retardo médio, e conseqüente transmissão do seu conteúdo. Com poucas alterações é possível incluir outras informações relevantes como:

- maior valor de retardo observado no período;
- menor valor de retardo observado no período;
- contadores de quadros recebidos, distribuídos em faixas de valores de retardo.

O último item, que fornece a distribuição dos retardos observados, oferece uma idéia mais precisa dos efeitos experimentados pelos quadros e tem especial relevância em testes onde o retardo sofre grandes variações. De posse desses valores, o responsável pelos testes poderá realizar cálculos estatísticos adequados, conforme a sua necessidade. Também é possível a evolução deste mecanismo de maneira a obter diversas médias de retardo ao longo de intervalos de tempo dentro do período. No momento da transmissão dos dados para o computador, seriam

enviados vários valores de retardo médio, correspondentes a frações do segundo anterior. Esses resultados também são importantes quando temos grandes variações de retardo, pois são capazes de apontar os instantes em que essas alterações ocorreram.

É de extrema importância ressaltar que todas essas evoluções descritas aumentam a demanda por recursos do *hardware* em uso. Quanto maior o número de intervalos de tempo ou intervalos de retardo a serem contabilizados, melhores serão os resultados, porém mais unidades lógicas do FPGA serão usadas para a implantação da lógica e armazenamento dos dados. A combinação das duas sugestões feitas logo acima permitiria a obtenção, não de apenas um valor médio de retardo por segundo, mas sim uma tabela relacionando o retardo e o tempo no último segundo.

Tempo(ms) Retardo(μ s)	0	200	400	600	800
< 1					
1 – 200					
201 – 400					
401 – 600					
601 – 800					
801 – 1000					
> 1000					

Tabela 1 – Modelo de matriz de resultados obtidos pelo módulo receptor, relacionando o retardo médio em faixas de tempo do segundo anterior.

A combinação de todas as tabelas, ao fim do teste, poderá ser uma rica fonte de informações para posterior processamento de cálculos estatísticos ou obtenção de gráficos detalhados.

4.5. Comentários e conclusões

Ao longo do presente capítulo é apresentada a arquitetura básica de um analisador de redes e as arquiteturas funcionais capazes de avaliar a máxima capacidade de transmissão, o retardo e o número de quadros perdidos, em uma rede de telecomunicações.

5 Realização do equipamento

Neste capítulo vamos detalhar todo o processo de desenvolvimento e realização do equipamento de teste. São descritos os equipamentos usados e a arquitetura adotada para a solução, e mencionados problemas que foram contornados ao longo do trabalho, além das limitações práticas encontradas.

5.1. Detalhes da tecnologia adotada

A primeira grande decisão de um projeto envolvendo tecnologia FPGA está exatamente na escolha dos componentes que serão usados. Além dos *chips*, que existem em inúmeras versões, outras peças fundamentais à solução final devem ser definidas quanto a sua existência e quantidade. Memórias, portas de comunicação de diferentes padrões, frequência de operação, conversores A/D são elementos fundamentais no desenvolvimento de um projeto. O principal entrave está no fato de que não é fácil estimar no início de um projeto os requisitos exatos que serão necessários. Mesmo desenvolvedores com grande experiência podem preferir não precisar todos os requisitos. Exatamente para diminuir os impactos dessa realidade, inúmeros fabricantes vendem placas chamadas de “desenvolvimento”. São placas, às vezes de alto custo, que possuem diversos recursos que possam vir a ser necessários no desenvolvimento do projeto. Ainda assim, existem variações de placas de desenvolvimento com diferentes características e recursos, o que obriga os desenvolvedores a ter uma mínima idéia da dimensão do seu projeto. Geralmente, opta-se por iniciar os trabalhos em placas com recursos superdimensionados aos quais se acredita precisar. O aparente desperdício será convertido em maior maleabilidade ao longo do trabalho e permitirá a otimização do produto final para implantação em *hardware* adequado.

Neste trabalho foi usada uma placa de desenvolvimento da empresa Xilinx, modelo ML405. Dentre as principais características dessa placa, podemos citar:

- *chip* FPGA fixo, modelo XC4VFX20-FF668, família Virtex4[11]. Este *chip* possui um processador Power-PC embarcado, permitindo execução simultânea de programas. Possui ainda 4 blocos DCM (*Digital Clock Manager*) e 8 MGT (*Multi-Gigabit Transceiver*) que serão importantes neste trabalho e posteriormente melhor explicados;
- entrada de cartão de memória CompactFlash, para transporte de programação;
- porta serial UART, RS-232[12];
- 256 MB de memória tipo DDR2 DIMM;
- 64 MB de memória DDR SDRAM;
- saída Ethernet em cobre, padrão 1000Base-T;
- conector SFP para instalação de transceptores de cobre ou fibra-óptica;
- conectores comuns em padrões de computadores pessoais, ideal para este desenvolvimento, como USB, SATA, PS/2 e VGA.

As placas de desenvolvimento permitem aos desenvolvedores trabalhar com maior flexibilidade e menos restrições. Em trabalhos que envolvem a prova de conceitos, são muito importantes: podem ser as principais responsáveis pela viabilidade do desenvolvimento. A criação de uma placa, ou seja, projeto e montagem do protótipo, é um processo bastante caro e não pode ser repetido inúmeras vezes ao longo do desenvolvimento de um projeto, sob pena de exigir um orçamento elevadíssimo. Idealmente, o produto final é montado apenas quando praticamente todos os trabalhos e estudos envolvidos já estão concluídos. Ainda assim, é prática comum dos fabricantes produzir placas prevendo alterações futuras ou que possam servir a várias versões e modelos futuros daquele produto.

5.1.1. O Processador embarcado Power-PC

O *chip* FPGA disponível possui embutido um microprocessador PowerPC, fisicamente construído, modelo 405 de 32 bits, desenvolvido pela IBM e voltado

para dispositivos embarcados SoC (*System-on-a-chip*)[13], sendo capaz de operar até 450 MHz.

A principal vantagem do processador está na possibilidade de incorporar lógicas em programas (*software*) à solução. Já foi mencionado que a lógica implantada no *hardware* FPGA é sempre limitada pela capacidade do *chip*, e que lógicas mais complexas ocupam mais blocos do FPGA, mas deve-se ter ciência de que as lógicas executadas em programa levam mais tempo para fornecer os resultados. Uma dúvida comum que pode surgir é de que, se optamos por um FPGA porque precisamos de alto desempenho, qual seria a vantagem de transferirmos parte do trabalho para um mecanismo que oferecerá baixo desempenho? A resposta está no fato de que, na maioria das aplicações, nem todas as lógicas que a compõem, precisam de alto desempenho. O processador nos oferece a oportunidade de transferir lógicas complexas e com menor exigência de desempenho, para programas que serão executados pelo processador. Cabe ressaltar que aplicações integralmente baseadas em *software* costumam ter custos muito inferiores.

Por exemplo, imagine que a um determinado equipamento esteja conectada uma tela de cristal líquido, responsável por exibir ao operador informações básicas sobre o andamento do trabalho. Telas, em geral, possuem tempo de atualização na ordem de milissegundos e, se atualizássemos as informações exibidas a esta velocidade, seria impossível a um ser humano acompanhá-las. A tarefa de montar informações para exibição em telas pode ser complexa, por envolver a montagem e alinhamento de textos ou geração de imagens, exigindo grande esforço para ser realizada e desenvolvida em lógica FPGA. No entanto, por possuir baixa exigência de velocidade, em comparação ao desempenho de lógica FPGA, pode ser realizada no processador.

Neste trabalho, as tarefas básicas de controle e comunicação com um microcomputador externo, serão realizadas por programas sendo executados no processador. As ferramentas de desenvolvimento da Xilinx que serão usadas possuem compilador cruzado para o processador PowerPC, na linguagem C/C++.

5.1.2. DCM - *Digital Clock Manager*

O DCM[14] é um gerenciador digital de *clock*, capaz de operar como sintetizador flexível de frequência, por multiplicação e/ou divisão, e produzir saídas com defasagem configurável. Todo circuito digital, baseado em FPGA, deve possuir um *clock* próprio, normalmente através de cristal, em frequência fixa. É possível, no entanto, que diferentes componentes do circuito trabalhem em frequências distintas e, no caso de um *chip* FPGA, isso pode ocorrer entre blocos implantados dentro do mesmo *chip*. Como a introdução de outros cristais para a criação de outros *clocks* aumenta o custo e a complexidade do projeto, o DCM se mostra como uma excelente saída para a obtenção destes sinais, sem a necessidade de alterações no projeto de construção do circuito. O DCM também pode ser usado para permitir a comunicação com interfaces externas como memórias, portas de comunicação ou unidades de disco, em suas frequências específicas de trabalho.

O funcionamento básico de um bloco DCM se dá pela injeção de um sinal de referência, conhecido, e uma de suas saídas em uma porta específica de referência. É essa realimentação que permitirá ao bloco funcionar com estabilidade.

A figura abaixo representa um módulo DCM típico:

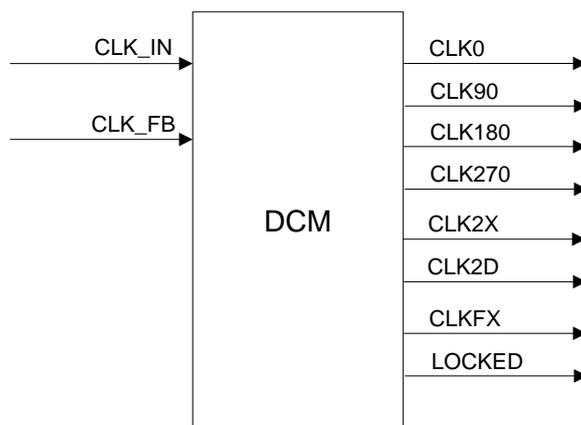


Figura 10 – Bloco DCM

- CLK_IN: Entrada de sinal de *clock*;
- CLK_FB: Entrada de referência. Pode ser ligado às portas CLK0 ou CLK2X dependendo do objetivo desejado;
- CLK0: Saída com a mesma frequência da entrada;
- CLK90: Saída com desvio de 90° em relação a CLK0;
- CLK180: Saída com desvio de 180° em relação a CLK0;
- CLK270: Saída com desvio de 270° em relação a CLK0;
- CLK2X: Saída com a mesma fase e dobro da frequência de CLK0;
- CLK2D: Saída com a mesma fase e a metade da frequência de CLK0;
- CLKFX: Saída com frequência sintetizada, conforme multiplicador e divisor programados. Mesma fase de CLK0;
- LOCKED: Indica que os sinais de saída estão estáveis.

5.1.3. MGT - Multi-Gigabit Transceiver

O bloco MGT[15] é um componente capaz de transmitir e receber dados em altas velocidades visando comunicação através de diversos padrões de redes e dispositivos de armazenamento. O MGT disponível na família Virtex-4, chamado de MGT Rocket-IO Transceiver, pode operar entre 622 Mb/s e 6,5 Gb/s, faixa na qual trabalham vários padrões, entre eles:

Padrão	Taxa (Gb/s)
Gigabit Ethernet	1,25
SONET OC-12	0,622
Fibre-Channel	1,06 / 2,12 / 4,25
PCI Express	2,5
Serial ATA	1,5 / 3,0

Tabela 2 – Velocidades de operação de alguns padrões de comunicação, que podem ser obtidas com o uso do MGT

O MGT ainda é capaz de codificar os dados conforme padrões comuns e pré-definidos, ou programados em lógica FPGA. Será de especial interesse para esta dissertação o padrão 8B/10B[2] usado nas aplicações Gigabit-Ethernet.

5.1.4. PLB – Processor Local Bus

No capítulo 2 foi apresentado o barramento OPB, ferramenta responsável por permitir a comunicação entre diversos dispositivos ou módulos implantados no *chip* FPGA. Conforme foi dito, o OPB não possui alta capacidade de transferência de dados, sendo, portanto, inadequado para dispositivos que requeiram trabalhar em alta velocidade. O exemplo mais comum é o processador que precisa de respostas o mais curtas possível ao acessar ou gravar dados em memória RAM. Para atender especialmente esta necessidade, foi desenvolvido um barramento de dados veloz, chamado de PLB[16].

O barramento PLB será usado por um grupo seletivo de dispositivos e é completamente independente de barramentos OPB. Para permitir a comunicação entre dispositivos conectados ao OPB e ao PLB, deverá ser usada uma ponte, que fará a tradução de endereçamento entre os dois barramentos. Um exemplo de uso da ponte é a possibilidade de permitir a um programa rodando no processador, buscar ou enviar informações para módulos conectados no OPB. A figura abaixo demonstra o uso da ponte PLB-OPB[17].

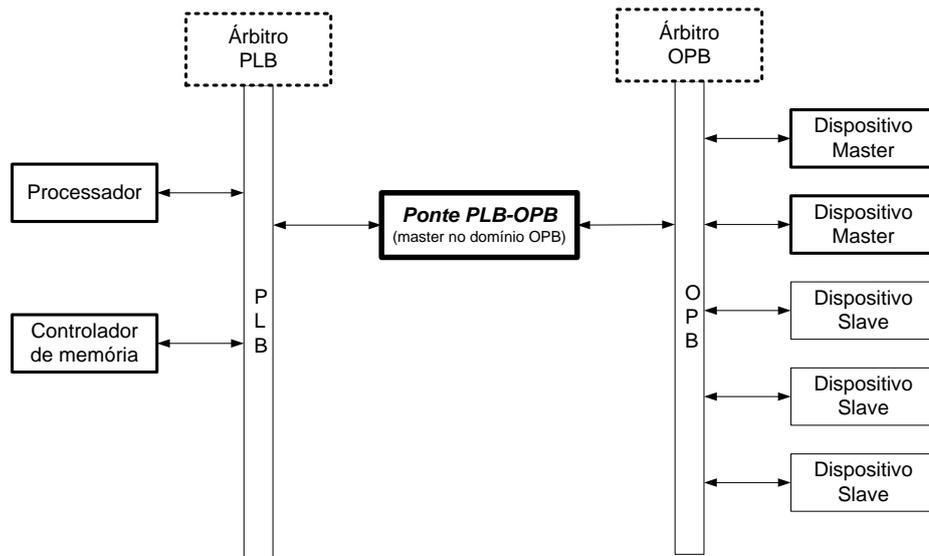


Figura 11 – Diagrama dos barramentos PLB e OPB, e a ponte que permite a comunicação entre eles

5.2. Módulos

A partir deste ponto, vamos começar a detalhar tecnicamente todo o desenvolvimento da solução, desde a lógica inicial até o primeiro protótipo. No capítulo 4 foi apresentada uma curta idéia da organização que seria adotada neste projeto, como base para a compreensão da lógica de trabalho descrita na seqüência.

Conforme foi mencionado, o equipamento de testes será composto de, basicamente, 3 módulos: transmissão, recepção e controle. O funcionamento básico de cada um deles será descrito em detalhes, assim como suas conexões.

5.2.1. Módulo de transmissão

É responsável por gerar um número pré-determinado de quadros, todos com um mesmo tamanho fixo, também pré-determinado. O controle do módulo de transmissão é realizado pelo módulo controlador, através de uma série de sinais

entre ambos. Através destes sinais, que serão a seguir relacionados, o processo de transmissão é configurado e ativado. O diagrama abaixo representa o módulo transmissor relativo à primeira versão apresentada, capaz de avaliar apenas a perda de quadros.

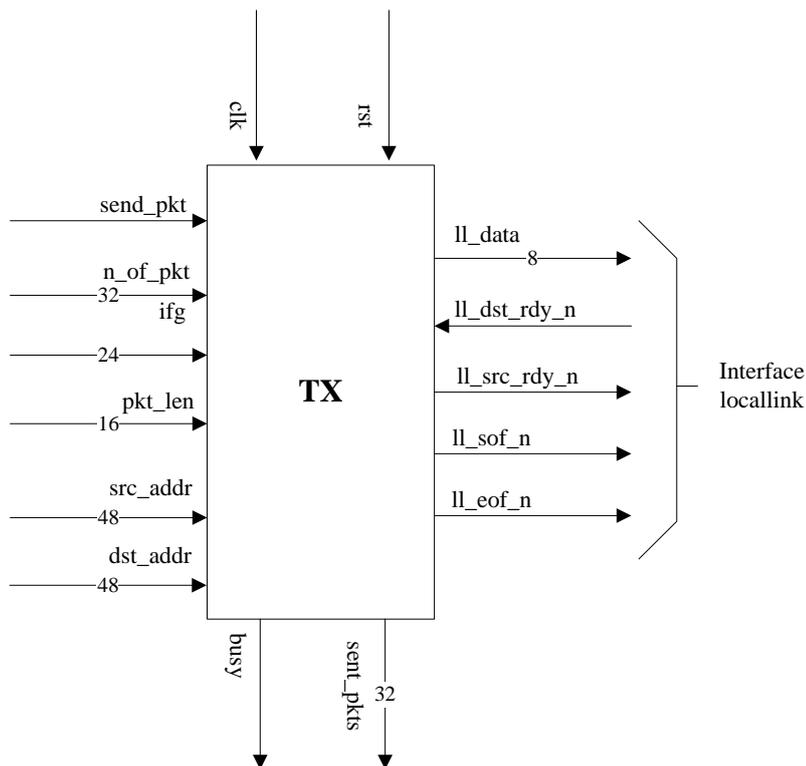


Figura 12 – Diagrama do módulo transmissor – primeira versão proposta

Sinais do módulo controlador para o módulo transmissor:

- `dst_addr` – 48 bits – Endereço MAC de destino que será inserido nos quadros;
- `src_addr` – 48 bits – Endereço MAC de origem que será inserido nos quadros. Na recepção, apenas os quadros com determinados endereços de origem e destino (`src_addr` e `dst_addr`) serão analisados;
- `pkt_len` – 16 bits – Tamanho fixo, em octetos, dos quadros transmitidos;

- IFG – 24 bits – Comprimento, em octetos, do intervalo entre o fim da transmissão de um quadro e o início do seguinte. É importante para permitir controle da taxa de transmissão desejada;
- n_of_pkt – 32 bits – Define o número de quadros que serão transmitidos no teste, ininterruptamente;
- send_pkt – 1 bit – Deve ser ligado pelo controlador durante 1 ciclo de CPU, o suficiente para o transmissor perceber sua existência e iniciar o processo de transmissão;

Sinais do módulo transmissor para o módulo controlador

- busy – 1 bit – Indica que há um teste em andamento;
- sent_pkts – 32 bits – Informação do número de quadros já transmitidos. Esse dado permite o acompanhamento do teste e é usado pelo módulo de controle no cálculo de estatísticas em tempo-real.

Sinais de uso comum: sinais que são compartilhados por todos, ou parte dos módulos da solução.

- Clk – 1 bit – Sinal de *clock*. É comum aos módulos de transmissão, recepção e controle de maneira que todos trabalhem sincronamente. Este sinal também alimenta as lógicas de rede em camadas inferiores, sendo a referência de tempo para a comunicação pela interface *local-link*;
- Rst – 1 bit – Sinal de ‘Reset’, compartilhado por todos os módulos da solução que, ao perceberem sua presença, retornam ao estado inicial, interrompendo qualquer operação em curso.

Conforme foi descrito no capítulo anterior, para possibilitar a avaliação do retardo dos quadros, será necessário incluir em cada quadro a informação do instante em que foi transmitido. Abaixo, o diagrama do bloco de

transmissão com a inclusão de uma entrada do relógio de onde será retirada essa nova informação, representada pelo sinal ‘delay_clk’, de 48 bits, em destaque:

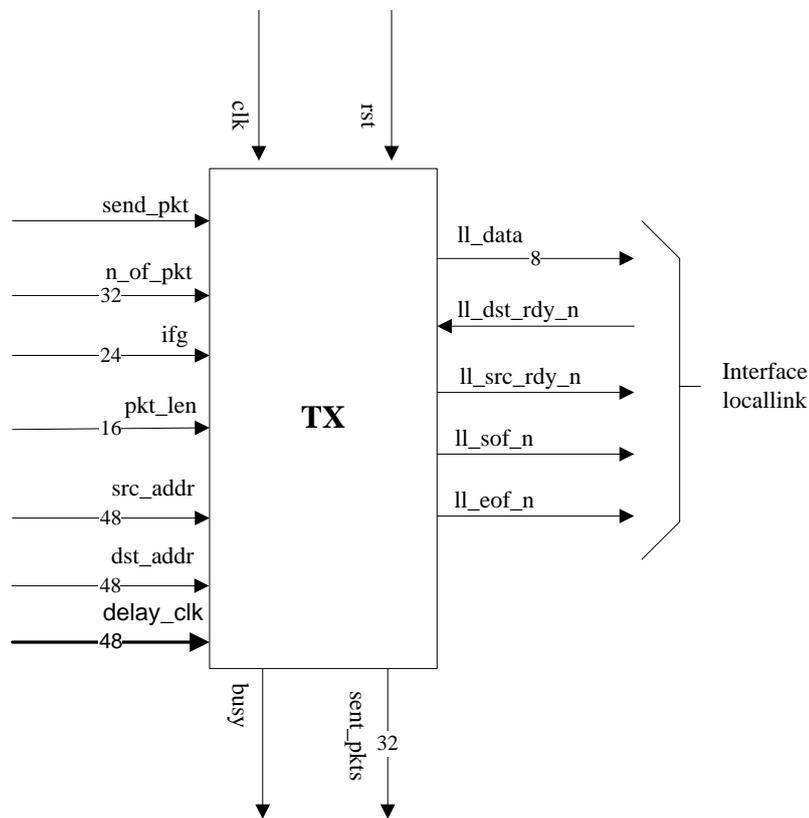


Figura 13 - Diagrama do módulo transmissor – segunda versão proposta, incluindo informações necessárias para o cálculo do retardo.

5.2.1.1. Cálculo do intervalo entre quadros (IFG)

O sinal IFG define o intervalo de tempo que o transmissor deve aguardar entre o término de um quadro e o início de outro. O IFG é o mecanismo que permite controlar a taxa de transmissão durante o teste. Uma vez que a velocidade de transmissão do GbE, ou velocidade de linha, é constante e igual a 1,25 Gbps, é necessário reduzir a taxa de transmissão de quadros para conseguirmos diminuir a taxa média de dados transmitidos.

A unidade que foi definida neste trabalho para o IFG equivale ao tempo de transmissão de um octeto no padrão Gigabit Ethernet. O motivo para esta escolha é devido ao fato de que, segundo o padrão, o intervalo entre dois quadros deve ser múltiplo de um octeto, ou byte.

Seja R a taxa de transmissão do Gigabit Ethernet

$$R = 1,25Gbps = 1,25 \times 10^9 \text{ bits} / s$$

Como o Gigabit Ethernet utiliza, em quase¹ todas as configurações, a codificação 8B/10[2] que transforma cada bloco de 8 bits em 10 bits, a taxa de dados transmitidos, excluindo-se o adicional devido à codificação, será:

$$R = \frac{1,25 \times 10^9}{10} = 1,25 \times 10^8 \text{ bytes} / s$$

Portanto, o tempo de transmissão de um byte de dados é:

$$T = \frac{1}{1,25 \times 10^8} = 8ns$$

A taxa de quadros transmitidos é:

$$FR = \frac{R}{\text{preâmbulo} + L + ifg} \text{ (quadros/s)}$$

Sendo:

- preâmbulo: limitador dos quadros Ethernet – 64 bits;
- L: tamanho dos quadros (supondo todos os quadros com o mesmo tamanho);
- IFG: 12 bytes (mínimo) * 8

O tamanho mínimo do IFG, definido pelo padrão, é de 12 bytes. A taxa de transmissão de quadros e dados será controlada pela variação do tamanho deste intervalo.

A taxa de transmissão de dados (R_d) pode então ser calculada como:

¹ Exceto em cabo de par trançado, padrão 1000Base-T

$$Rd = FR \times L$$

$$Rd = \frac{R}{preâmbulo + L + IFG} \times L$$

Dada a taxa de transmissão de dados desejada para o teste, calculamos o IFG, em octetos, a ser usado:

$$IFG = \left(\frac{R \times L}{Rd} \right) - preâmbulo - L$$

5.2.1.2. Capacidade máxima de transmissão

O item anterior dá uma informação sobre a capacidade de transmissão no padrão Ethernet: devido a existência do preâmbulo e do IFG mínimo, a máxima taxa de transmissão de dados, incluindo os cabeçalhos, poderá ser próxima, mas nunca igual à taxa líquida de bits transmitidos.

Quanto maior o tamanho dos quadros transmitidos, proporcionalmente menor será o efeito desses intervalos obrigatórios sobre a capacidade máxima de transmissão.

Como o IFG deve ter um valor mínimo de 12 octetos e o preâmbulo de 8 octetos, podemos calcular a máxima taxa de quadros e bits transmitidos para determinados tamanhos de quadro:

L (bytes)	Máx FR	Máx Mbps	Rel Tx Max (%)
64	1.488.095	761,90	76,2
128	844.595	864,86	86,5
256	452.899	927,54	92,8
512	234.962	962,41	96,2
768	158.629	974,62	97,5
1024	119.732	980,84	98,1
1280	96.154	984,62	98,5
1518	81.274	987,00	98,7

Tabela 3 – Máxima capacidade de transmissão[18], no padrão Gigabit-Ethernet, para quadros de diferentes tamanhos

Onde:

- L: tamanho (fixo) dos quadros transmitidos;
- FR: taxa de quadros transmitidos por segundo;
- Rel Tx: relação porcentual da máxima taxa de transmissão em relação à taxa de linha de 1 Gbps.

A relação entre a máxima taxa de transmissão, o IFG utilizado e o tamanho do quadro é expressa no gráfico a seguir, onde cada curva está associada à dimensão do quadro utilizado:

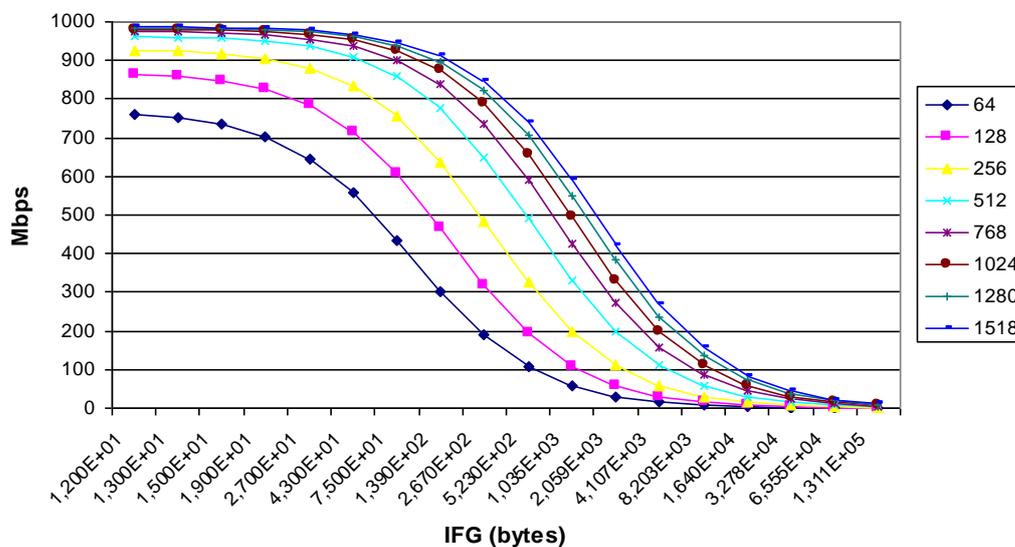


Figura 14 – Relação entre a taxa de transmissão e o IFG, para diferentes tamanhos de quadro

5.2.1.3. Fluxograma do módulo de transmissão

O fluxograma a seguir ilustra a lógica elaborada para o módulo transmissor e que foi utilizada na descrição da máquina de estados que coordena sua operação.

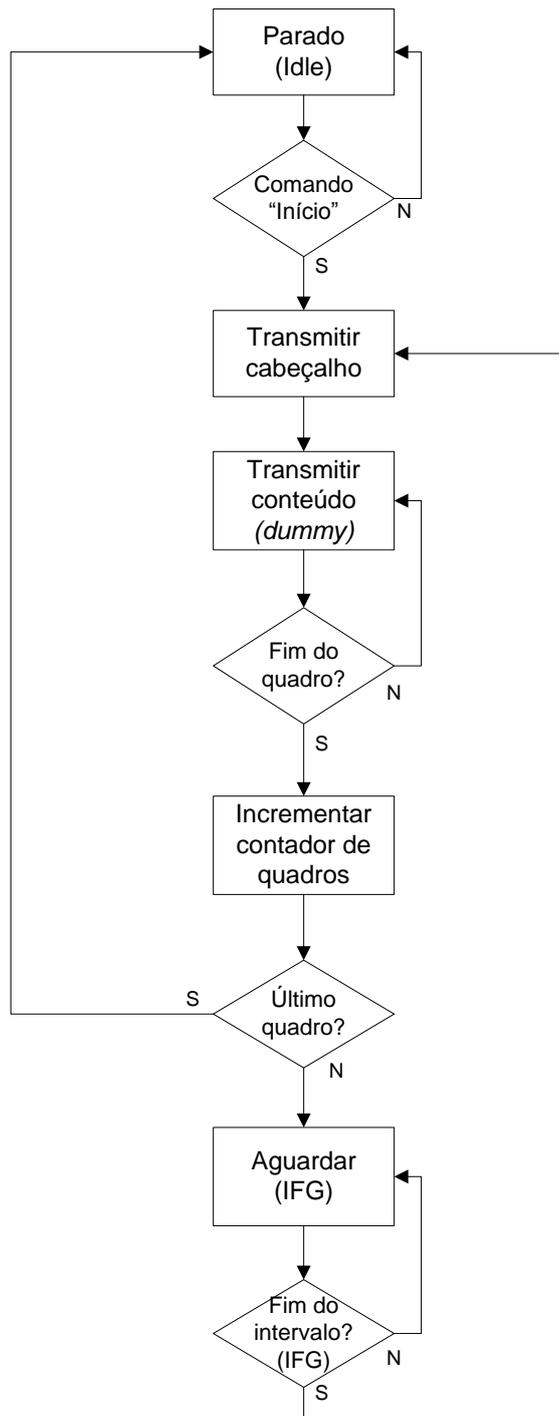


Figura 15 – Fluxograma da lógica do módulo transmissor, que descreve a sua máquina de estados

O conteúdo do quadro é composto, além dos cabeçalhos e identificadores relativos ao teste, de octetos para complementação do tamanho desejado. Esse preenchimento pode ser feito com bytes aleatórios ou seqüenciais. É de grande importância ressaltar que o módulo transmissor não é responsável pela montagem final dos quadros. Essa incumbência é designada às camadas inferiores do padrão

Ethernet, para quem o módulo transmissor entrega apenas uma parte do quadro final. Dentre as funções das camadas inferiores está o delineamento dos quadros, por meio do preâmbulo, o cálculo do código verificador de erro, CRC, e a codificação 8B/10B.

5.3. Módulo de recepção

O módulo de recepção é responsável por receber os quadros transmitidos e realizar as avaliações estatísticas iniciais. As duas versões mencionadas no capítulo 4 serão apresentadas. A primeira, apenas com a avaliação de quadros perdidos, e a segunda, capaz de também avaliar os retardos envolvidos.

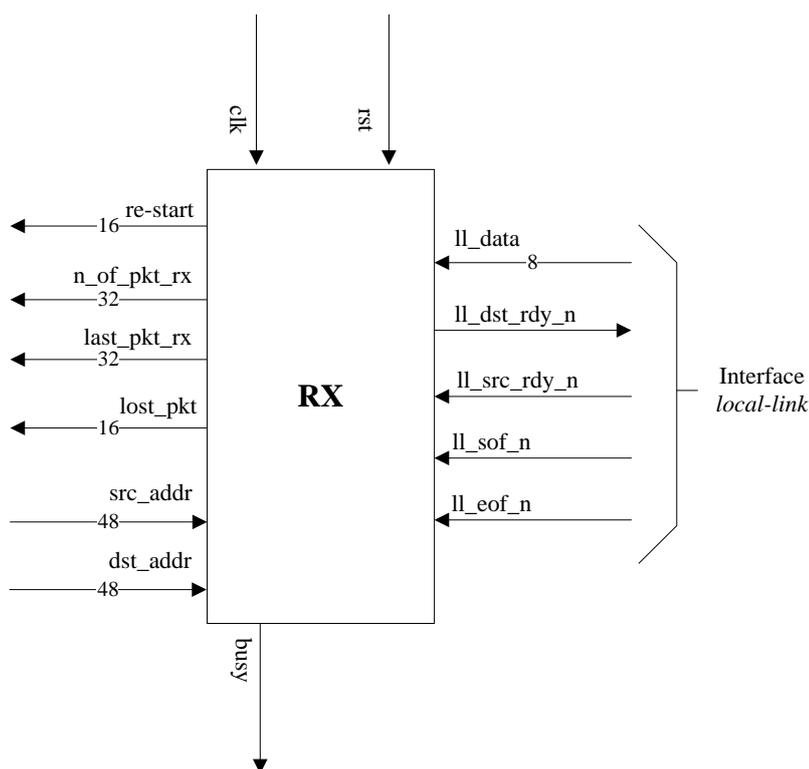


Figura 16 – Diagrama do módulo receptor – primeira versão proposta

Sinais do módulo controlador para o módulo receptor:

- `dst_addr`: 48 bits – endereço MAC de destino inserido nos quadros;

- **src_addr:** 48 bits – endereço MAC de origem inserido nos quadros. Apenas os quadros com os respectivos endereços de origem e destino (**src_addr** e **dst_addr**) são analisados;
- **Re-start** – 16 bits – Indica o número de testes que foram iniciados desde o último *reset*. Vinculado a um contador, é útil para permitir a identificação de eventuais problemas. Esse valor só pode sofrer incremento no início de um teste. Caso exista a ocorrência deste evento em outros momentos, pode indicar recebimento duplicado de quadros ou a presença de elementos interferentes na rede. Os procedimentos a adotar diante desses cenários variam conforme a necessidade e objetivos das pessoas responsáveis, sendo, o mais simples deles, a emissão de um alerta visual ou sonoro pelo software no computador;
- **n_of_pkt_rx:** 32 bits – indica o número de quadros recebidos desde o início do teste;
- **last_pkt_rx:** 32 bits – indica o número identificador do último quadro recebido até aquele instante;
- **lost_pkt:** 32 bits – indica o número de quadros perdidos, armazenado no acumulador correspondente;
- **busy:** 1 bit – indica que o módulo está processando um teste no presente momento. O sinal ‘busy’ é ativado quando o primeiro quadro, com a identificação adequada, é recebido; é desligado após o recebimento do último quadro ou um determinado tempo após o módulo transmissor tê-lo enviado. Esse intervalo de tempo é chamado de *timeout*.

Sinais de uso comum: sinais que são compartilhados por todos, ou parte dos módulos da solução.

- Clk: 1 bit – sinal de *clock*. É comum aos módulos de transmissão, recepção, controle e blocos de rede (MAC), de maneira que todos trabalhem sincronamente;
- Rst: 1 bit – sinal de *reset*, compartilhado por todos os módulos da solução que, ao perceberem sua presença, retornam ao estado inicial, interrompendo qualquer operação em curso.

A segunda versão do módulo de recepção foi desenvolvida conforme indicado pelo segundo modelo apresentado, e é capaz de aferir o tempo médio de retardo sofrido pelos quadros.

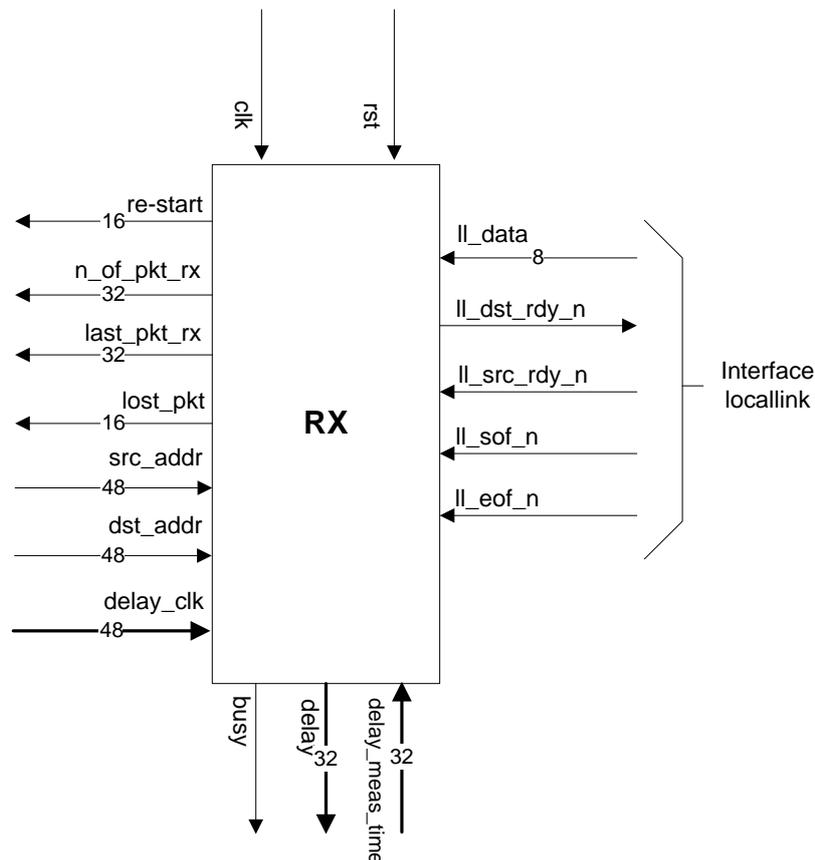


Figura 17 - Diagrama do módulo receptor – segunda versão proposta, capaz de calcular o retardo dos quadros

O diagrama do bloco é similar ao anterior, porém com a adição de novos sinais ligados ao módulo controlador. São eles:

- `delay_clk`: 48 bits – recebe o valor corrente do contador de tempo do teste, usado como base para a avaliação do retardo;
- `delay`: 32 bits – indica o retardo médio sofrido pelos quadros no último intervalo de tempo avaliado. A unidade é a mesma base de tempo usada pelo contador de tempo;
- `delay_meas_time`: 32 bits – define o intervalo de tempo que será usado pelo módulo, com base no relógio mestre, para a avaliação do retardo médio. Esse intervalo deve ser definido considerando-se o período com que as estatísticas do teste serão repassadas pelo módulo de controle ao computador.

5.3.1. Fluxograma do módulo de recepção

* Um ciclo de testes chega ao fim em duas situações: quando o último quadro da seqüência é recebido, ou após um tempo limite sem o recebimento de novos quadros *timeout*. Em ambos os casos, será necessário o cálculo do retardo médio em relação ao tempo decorrido desde a última avaliação. Na hipótese de término por *timeout*, o número de quadros restantes deverá ser somado a contagem de quadros perdidos. Lembrando que este valor é conhecido pelo módulo receptor devido à transmissão da numeração dos quadros em ordem decrescente. O último quadro de um teste é identificado quando a sua numeração for igual a zero.

** O trecho de lógica no interior do retângulo tracejado (figura 18) é específica da segunda versão do módulo receptor, capaz de avaliar o retardo médio dos quadros. A primeira versão é igual à segunda, excluindo-se apenas o conteúdo deste retângulo.

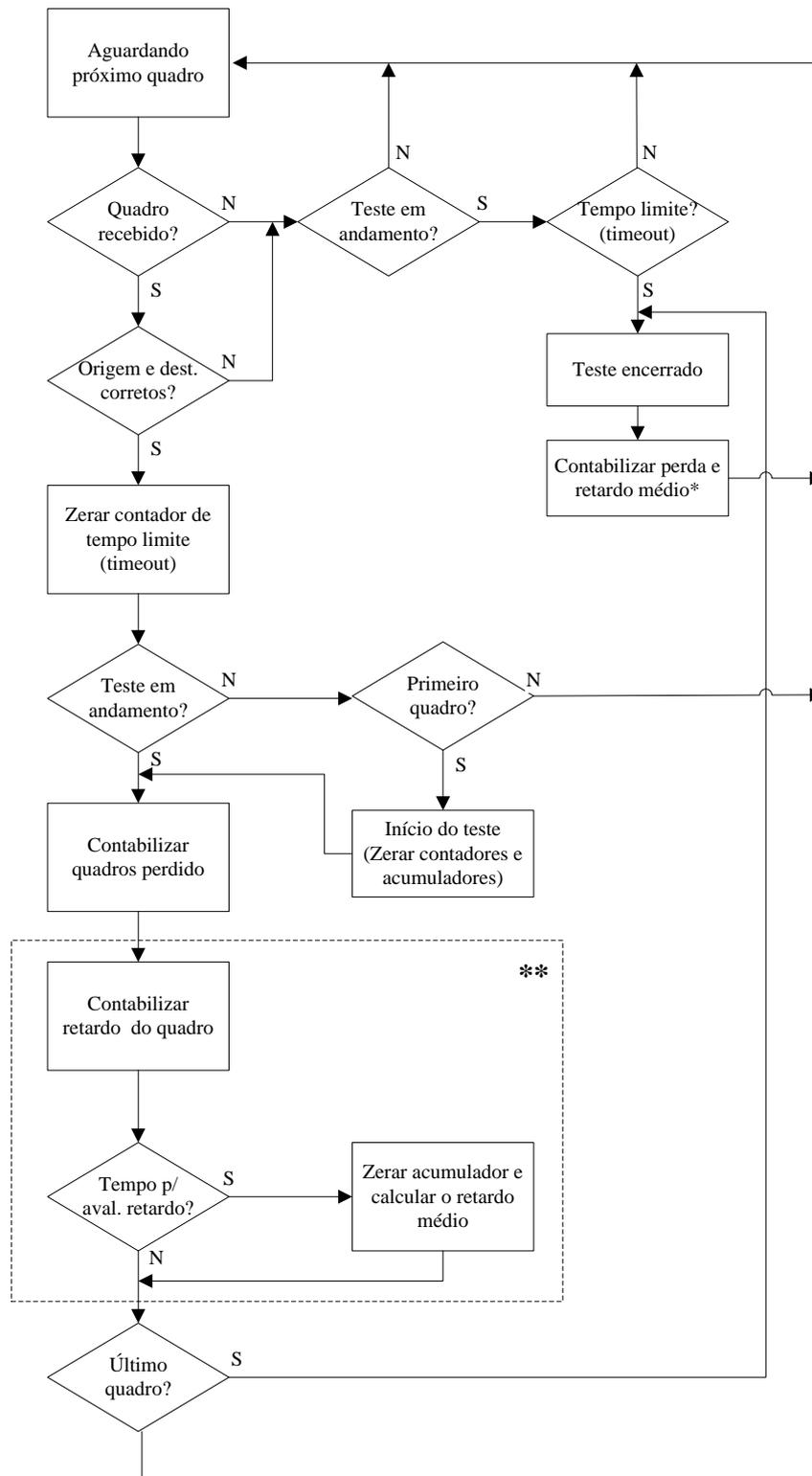


Figura 18 – Diagrama detalhado da lógica do módulo receptor, que descreve a sua máquina de estados

5.3.2. Envio de quadros para aprendizado

Antes de iniciar uma seqüência de testes, é importante ter certeza de que o dispositivo, ou dispositivos, ao longo do teste saberão encaminhar os quadros corretamente, até serem entregues ao módulo receptor. Para garantir este conhecimento, o módulo receptor deve transmitir, antes do início do teste, um quadro de aprendizado. O recebimento desse quadro pelos dispositivos resultará na atualização das suas tabelas de encaminhamento, de forma que estes saberão encaminhar corretamente os quadros destinados ao módulo receptor.

O quadro de aprendizado varia conforme o ambiente em teste. Em redes com o uso de IP, ele pode ser um pacote do tipo ARP[19], que informará a associação do endereço IP ao endereço MAC do receptor, além de ensinar aos elementos comutadores, como *switches*, a porta onde o receptor está conectado. No caso deste trabalho, em que focamos apenas o padrão Ethernet, sem nos preocupar com demais protocolos de camada superior, o quadro de aprendizado é mais simples, sendo apenas necessário transmitir um quadro tipo *broadcast*¹, sem necessidade de preenchimento com conteúdo específico. Certamente, a evolução deste equipamento exigirá mecanismos mais sofisticados e capazes de atender aos demais casos de aprendizado que possam ser necessários.

5.4. Controle e comunicação externa

Completando a solução, vamos descrever os mecanismos responsáveis pelo controle das funções básicas dos módulos de transmissão e recepção, e como a comunicação externa, com o microcomputador, é realizada. Essas tarefas estão distribuídas em dois módulos distintos que, comunicando-se entre si, permitem a transferência de comandos de operação e a apresentação dos resultados obtidos ao longo do teste. A primeira parte da lógica está implantada em um bloco de *hardware*, no FPGA, conectado diretamente aos módulos de transmissão e

¹ O quadro Ethernet, onde todos os bits do campo endereço de destino são '1', é chamado de *broadcast* e é encaminhado a todas as estações conectadas àquela rede.

recepção e operando sincronamente com eles. A segunda é composta por um programa sendo executado no microprocessador PowerPC. Como já foi mencionado anteriormente, a comunicação entre esses dois módulos é realizada através dos barramentos PLB e OPB, lembrando que a troca de informações entre os dois se dá através de uma ponte PLB-OBP.

A figura abaixo ilustra o diagrama do bloco de controle no circuito FPGA.

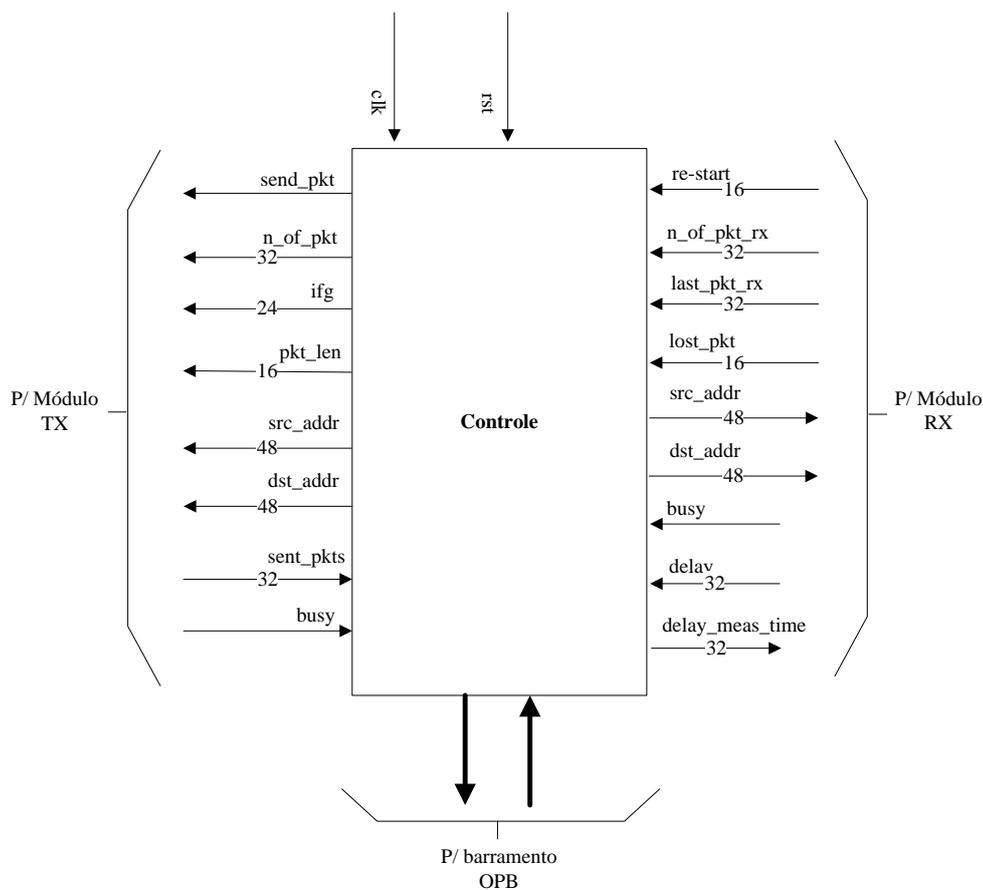


Figura 19 - Diagrama do módulo de controle

Um detalhe de extrema importância está no fato de que o sinal de *clock* é o mesmo para os três módulos: de controle, transmissor e receptor. O sincronismo entre os três é necessário para garantir a integridade das informações propagadas nos sinais entre eles. Se as frequências fossem distintas, ou se os sinais estivessem defasados, uma informação poderia ser lida durante o período em que estava sendo atualizada, ou seja, seria obtido um valor incorreto.

O barramento OPB, por definição, utiliza um sinal próprio de *clock*. Apesar de ser possível injetar o mesmo sinal já compartilhado pelos outros módulos, devemos lembrar que a estrutura criada para a implantação do barramento possui uma considerável complexidade. O sinal de *clock* usado no processo de transmissão e recepção deste trabalho visa a operação em uma taxa elevada, e utilizá-lo para alimentar o barramento OPB criará grandes restrições ao tempo máximo de propagação do sinal através das linhas de roteamento do FPGA. Por isso optamos, neste trabalho, por alimentar o barramento OPB com a metade da frequência do *clock* original.

O sinal de *clock* básico de 125 MHz, gerado por um cristal, foi dividido por dois, com o uso de um bloco DCM, e só então entregue ao barramento OPB. Este sinal é obrigatoriamente usado em todas as comunicações com o barramento, que trabalha de maneira síncrona. Pelo barramento OPB chegam requisições, que podem ser comandos ou pedidos de informação, conforme protocolo definido na implementação.

5.4.1. Isolamento do sinal de *clock* do barramento OPB

Os sinais que ativam o OPB e o *local-link* possuem frequências distintas ou defasadas. No presente trabalho, a frequência do sinal OPB resultará da divisão por dois da frequência do *local-link*, estando estes sinais alinhados em fase. Desta forma, será garantido que uma determinada transição positiva de referência sempre ocorrerá no sinal de menor frequência junto com o sinal de frequência superior. Como consequência, existe a garantia de integridade dos dados disponíveis nos sinais neste instante. Uma solução mais segura, capaz de trabalhar independentemente dos sinais de *clock*, envolve o uso de registradores intermediários e um semáforo responsável por sinalizar a disponibilidade das informações.

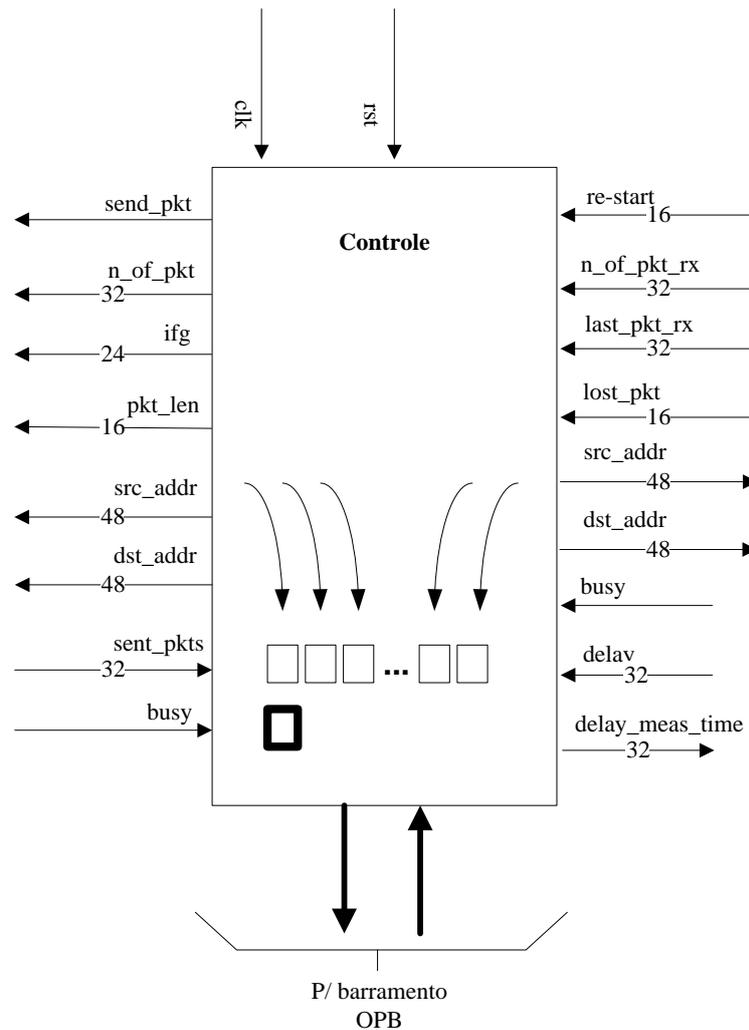


Figura 20 – Diagrama do mecanismo de troca de informações entre pontos alimentados por sinais de *clock* distintos

No diagrama acima, uma evolução do anteriormente apresentado, os blocos em cinza no interior do módulo representam esses registradores intermediários, enquanto o bloco em destaque representa o semáforo de disponibilidade dos dados. O funcionamento é bastante simples: os dados nesses blocos são atualizados em períodos referentes a ‘n’ ciclos do sinal de *clock*, ‘clk’ do módulo. Durante esse período o semáforo é mantido ligado, de forma a indicar que os dados nesses registradores estão disponíveis. Após ‘n’ ciclos, o estado do semáforo é alterado, de forma a sinalizar que os dados não estão disponíveis, e então as informações nos registradores são atualizadas. Durante este período, qualquer solicitação de leitura recebida pelo OPB será aceita, mas a resposta só poderá ser enviada quando o semáforo indicar a disponibilidade da informação. O

término da atualização dos dados nesses registradores é indicado pela mudança no semáforo.

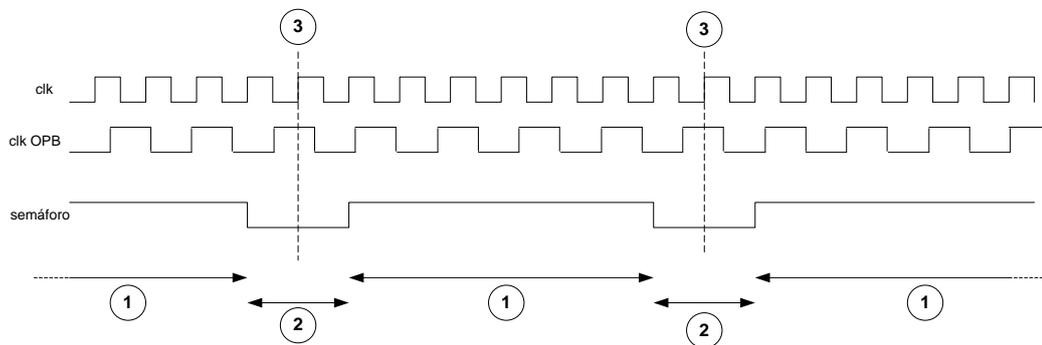


Figura 21 – Detalhe do funcionamento do semáforo proposto

Onde o intervalo:

1: representa o período em que a informação está disponível para leitura pelo dispositivo OPB;

2: representa o período em que a informação não está disponível. Se alguma requisição de leitura for recebida nesse intervalo, será necessário aguardar que o semáforo volte a indicar a disponibilidade dos dados;

3: representa o instante em que os registradores são atualizados com as informações enviadas pelos módulos de transmissão e recepção.

Uma consequência imediata deste mecanismo é a existência de um período mínimo necessário entre duas operações de leitura consecutivas. No entanto, devemos ressaltar que este tempo ainda é muito inferior ao requerido pelos demais processos envolvidos em uma operação de leitura ou escrita pelo barramento OPB e pela interface serial.

5.4.2. A gerência através de *software*

A segunda parte das operações de controle e gerência é realizada através de um programa que é constantemente executado pelo processador PowerPC. A este programa é designada a função de estabelecer a interface entre o módulo de

controle, em *hardware*, e o programa de administração, em execução em um microcomputador externo, conectado através de uma porta serial. O programa tem como funções principais receber comandos e solicitações de informações, além de traduzir dados brutos, obtidos do módulo de controle, ou alternar representações numéricas.

A principal vantagem no uso do programa é a simplicidade técnica que ele proporciona, se comparado ao trabalho necessário para a mesma realização em blocos de *hardware*. A comunicação serial, através de uma porta UART já existente, a leitura de comandos e a montagem de seqüências de caracteres como resposta, são exemplos dessas operações. Uma exploração mais aprofundada deste recurso possibilita ainda uma infinidade de tarefas de processamento de resultados de grande complexidade.

5.5.

Programa de gerenciamento e interface com o usuário

No extremo oposto da solução, um programa sendo executado em um microcomputador externo é responsável por duas importantes funções:

- 1: permitir a interação do usuário com todo o equipamento, repassando comandos;
- 2: exibir os resultados obtidos ao longo do teste.

O programa que será mostrado a seguir consiste em uma interface básica de testes, onde o usuário deve definir os parâmetros desejados e pode visualizar os resultados obtidos. Foi desenvolvido na linguagem C#[20] com o uso do ambiente “Microsoft Visual Studio 2008 Express”, na plataforma “Microsoft .NET”® versão 3.5, em modo gráfico (*winform*) para ambientes “Microsoft Windows”®. A escolha desta plataforma se deu pela sua simplicidade ao longo do desenvolvimento deste projeto. Versões em outras plataformas, inclusive independentes do sistema operacional, podem ser desenvolvidas com grande facilidade.

A imagem abaixo mostra a tela principal do programa:

Figura 22 – Tela principal do programa de administração

Na parte superior da janela, o usuário deve preencher os parâmetros desejados para o teste:

- a quantidade de quadros que será transmitida ininterruptamente ao longo do teste;
 - mínimo: 1
 - máximo: 2^{32}
- o tamanho fixo dos quadros, em bytes;
 - mínimo: 64
 - máximo: 1518
- taxa de transmissão, em Mbps;
 - mínimo: 0,1
 - máximo: 1000

Obs: São admitidas até 2 casas decimais
- endereços MAC de origem e destino.
 - Formatados em 6 octetos com representação hexadecimal e separados por ‘dois-pontos’.

Para iniciar o teste o usuário precisa apenas acionar o botão ‘Iniciar’.

Ao longo do teste, os resultados parciais são exibidos, como na imagem abaixo.

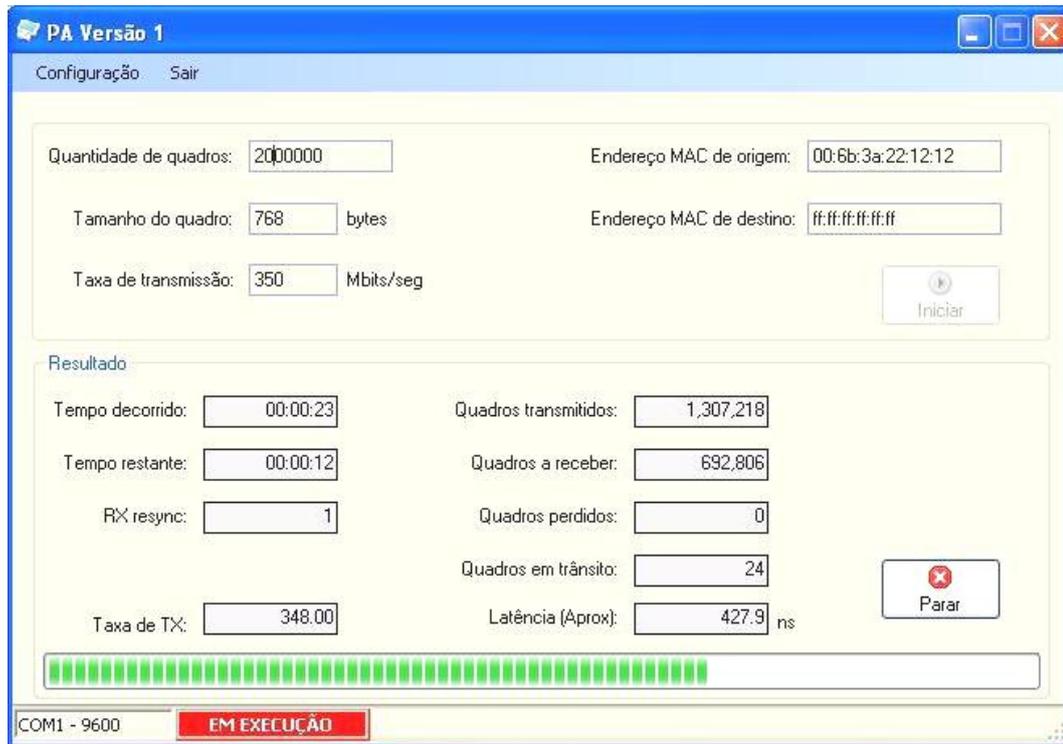


Figura 23 - Tela principal do programa de administração, exibindo resultados parciais, durante um teste

Duas telas adicionais de configuração fazem parte do programa. A primeira delas permite a configuração da comunicação pela porta serial do microcomputador:

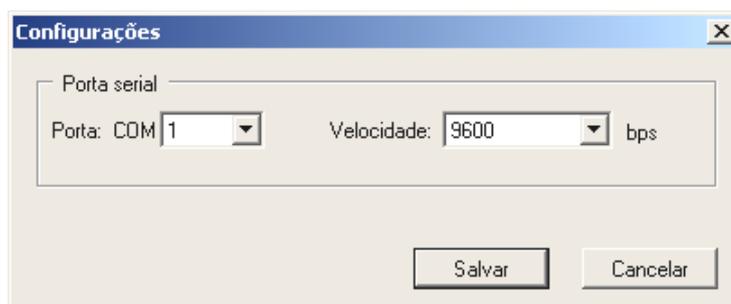


Figura 24 – Tela de configuração do acesso pela porta serial

A segunda oferece a opção de ajuste do período entre atualizações de resultados ao longo dos testes:

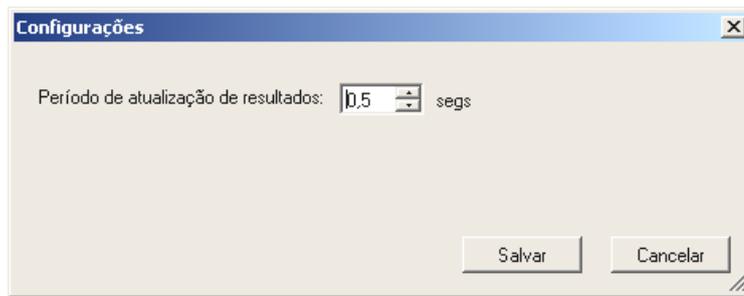


Figura 25 – Tela de configuração do período entre atualizações durante a realização de um teste

A configuração acima instrui o programa a emitir um comando para obtenção de resultados parciais do teste a cada meio segundo. Ao obter as respostas, as informações na tela são atualizadas.

5.6. Comentários e conclusões

Ao longo do presente capítulo foram apresentadas as principais etapas da realização do equipamento proposto. Inicialmente foram introduzidas as características da placa de desenvolvimento utilizada, destacando detalhes do processador Power-PC embarcado, do gerenciador de *clock* (DCM), do transceptor *Multi-Gigabit* (MGT) e do barramento PLB. Em seguida, os detalhes técnicos da solução adotada, incluindo os módulos de transmissão, recepção e controle, foram apresentados. Também foram detalhadas questões relativas ao cálculo do intervalo entre quadros e à avaliação da capacidade máxima de transmissão. Finalmente, foram apresentados os fluxogramas dos módulos desenvolvidos.

6

Testes realizados

Neste capítulo serão descritos os testes realizados com o equipamento e a posterior avaliação dos resultados obtidos. Estes, por sua vez, serão comparados aos resultados fornecidos por um equipamento comercial, quando submetido a um teste sob as mesmas condições.

6.1.

Descrição do teste

Os testes realizados para este trabalho objetivam a determinação da máxima capacidade de transmissão de um determinado elemento de rede, seguindo o algoritmo descrito no capítulo 2.2.4 deste trabalho, para quadros de diversos tamanhos. A decisão de determinados parâmetros do teste seguirá recomendações contidas na RFC 2544[21], que trata especificamente de procedimentos de testes, porém com maior abrangência em relação à proposta desta dissertação.

Seguindo recomendação da RFC 2544, item 9.1, neste teste serão utilizados quadros dos seguintes tamanhos, em bytes: 64, 128, 256, 512, 768, 1024, 1280. Outra questão importante diz respeito à quantidade de quadros transmitidos em cada passo do teste. Seguindo recomendação da RFC 2544, item 24, o número de quadros é calculado para que o teste dure 60 segundos.

6.2.

Descrição do elemento de rede escolhido

Uma questão fundamental para o teste envolve a escolha do elemento que terá seu desempenho avaliado. É importante que este equipamento possua capacidade máxima inferior ao limite disponível para o teste. Desta forma temos a garantia de que será possível submetê-lo a uma carga de trabalho excessiva e, conseqüentemente, encontrar seu ponto limite de operação sob as condições desejadas. Outra exigência imposta por este trabalho é a de que todo o teste será

realizado exclusivamente por portas ópticas, excluindo-se, portanto, o uso de portas e cabos de cobre.

O elemento de teste escolhido foi um sistema composto de um switch e um microcomputador, cujas características estão descritas abaixo, configurado para trabalhar em modo *bridge*, trafegando quadros entre duas interfaces.

- processador: Intel™ Pentium®-III, 750 MHz;
- interfaces de rede: Marvel™ Yukon, saída óptica com conector SC, multimodo 850nm;
- sistema operacional: Windows XP c/ SP2;
- modo bridge: nativo do sistema operacional, englobando apenas as duas placas utilizadas no teste;
- outras observações: foram desinstalados ou removidos todos os serviços e processos do sistema operacional não essenciais para a realização do teste. Desta forma reduz-se a concorrência por processamento, que poderia dificultar os testes ou produzir resultados imprecisos;
- *Switch* D-link DSG-1248T, gerenciável, com 48 portas para cabo de cobre par-trançado, em 10, 100 e 1000 Mbps.

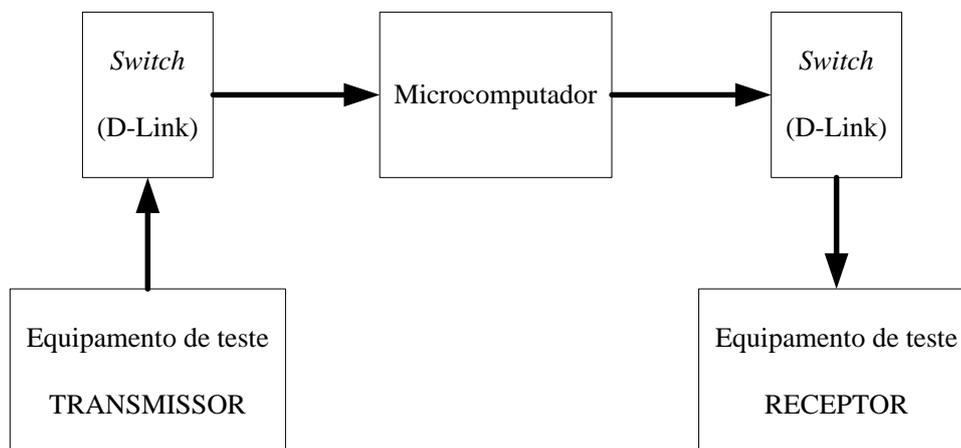


Figura 26 – Diagrama de blocos da montagem, para a realização dos testes

O *switch* foi posicionado entre os dispositivos para evitar uma situação, eventualmente verificada, de perda de comunicação entre o equipamento de referência e o computador, durante a realização de testes com carga

elevada. O *switch* foi previamente analisado, com o equipamento de referência, sendo comprovada a sua elevada capacidade de transmissão. Portanto, é possível concluir *a priori* que a presença do *switch* exercerá pouca, ou desprezível, influência sobre os testes, especialmente em relação à perda de quadros.

6.3. Teste de referência

Para avaliar a eficiência do equipamento proposto nesta dissertação, seus resultados serão comparados aos de um equipamento comercial, com proposta semelhante. Ambos os testes serão realizados sob, exatamente, as mesmas condições, e com os mesmos parâmetros de configuração, permitindo assim uma comparação mais precisa entre os resultados.

Equipamento escolhido:

- fabricante: Agilent Technologies;
- modelo: Framescope Pro®, N2620A;
- porta óptica: módulo SFP, conector LC, multimodo 850nm;
- Custo: US\$ 16.000 (duas unidades).

6.4. Resultados obtidos

Para evitar qualquer efeito causado por instabilidade do elemento em avaliação, os testes foram conduzidos no mesmo dia e repetidos 3 vezes, sendo utilizada a média entre eles como resultado final. Os raros resultados com elevada discrepância em relação aos demais foram descartados.

Na tabela abaixo, os resultados de máxima capacidade de transmissão são representados em ‘quadros por segundo’ de ambos os testes. O equipamento identificado como ‘I’, representa o trabalho desta dissertação. O equipamento identificado como ‘II’, representa o ‘Framescope Pro’, da Agilent Technologies.

Tamanho do quadro (Bytes)	Equipamento I (Max quadros/s)	Equipamento II (Máx quadros/s)	Dif % entre os equips II e I
64	103.125	104.177	1,020
128	99.316	100.553	1,246
256	93.750	93.334	-0,444
512	84.137	84.137	0,000
768	60.514	61.396	1,458
1024	46.630	47.186	1,192
1280	38.437	38.934	1,293
1518	33.058	33.243	0,560

Tabela 4 – Resultado dos testes, representando máxima capacidade de transmissão em quadros por segundo

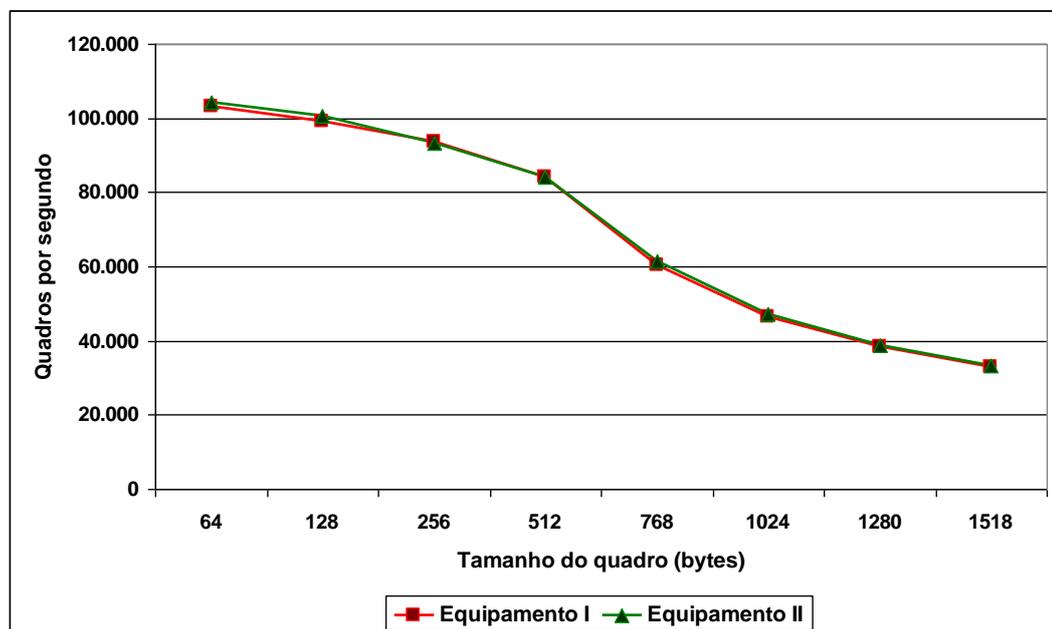


Figura 27 – Representação do resultado em forma gráfica, relacionando a máxima capacidade de transmissão, em quadros por segundo, com o tamanho dos quadros

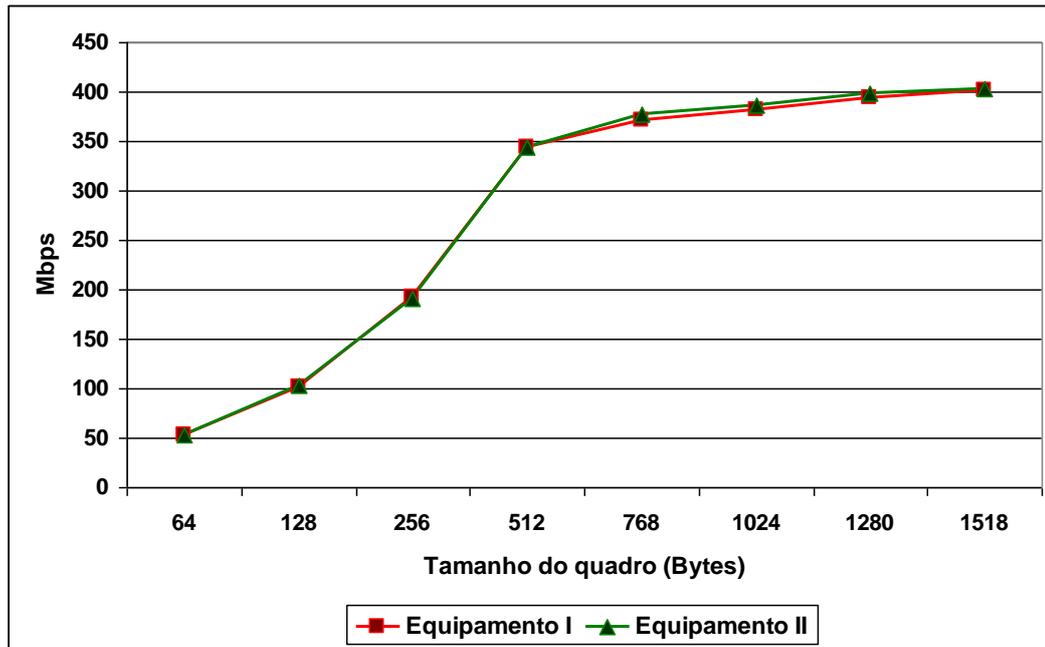


Figura 28 – Representação do resultado em forma gráfica, relacionando a máxima capacidade de transmissão, em taxa de bits por segundo, com o tamanho dos quadros

Nos quatro gráficos a seguir, apenas para fins de comparação, são apresentados os mesmos resultados acompanhados da máxima capacidade¹ teórica de transmissão em quadros e em bits por segundo, nas escalas linear e logarítmica.

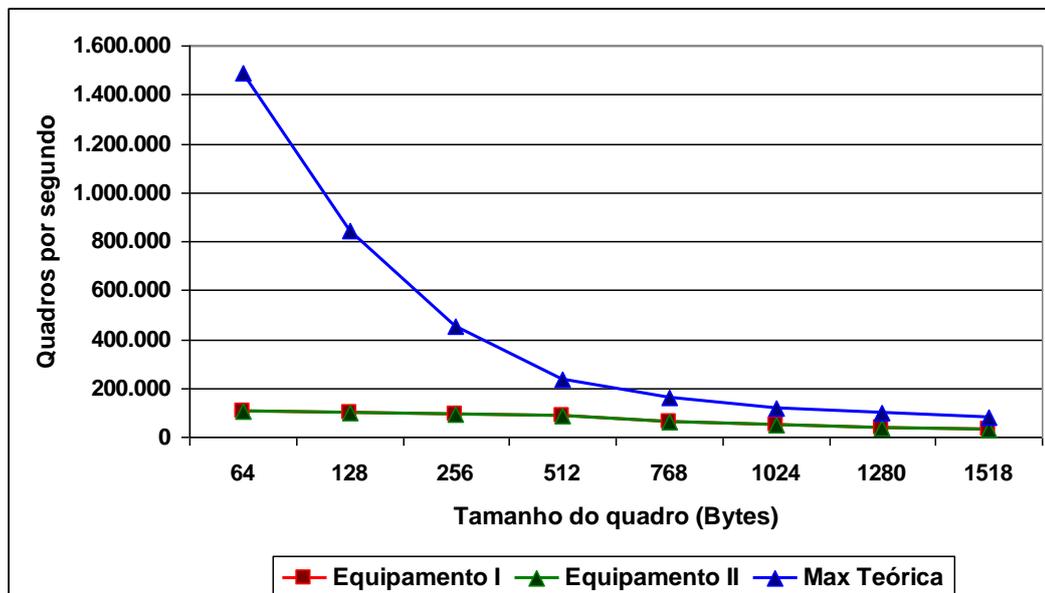


Figura 29 – Representação gráfica dos resultados em comparação com os limites teóricos, em taxa de quadros e escala linear

¹ Conforme apresentado no capítulo 2.2.4

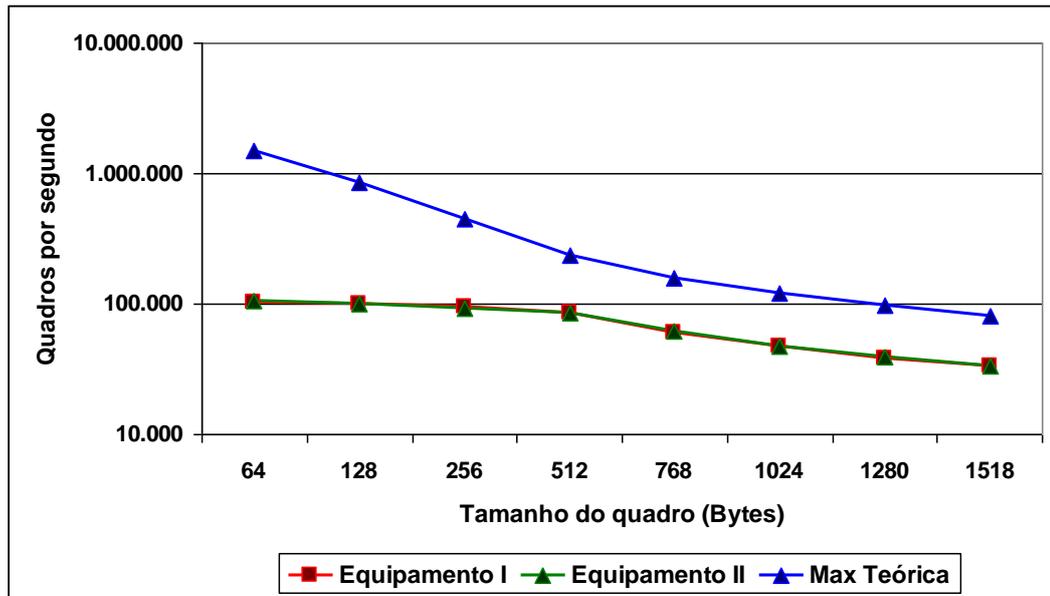


Figura 30 - Representação gráfica dos resultados em comparação com os limites teóricos, em taxa de quadros e escala logarítmica

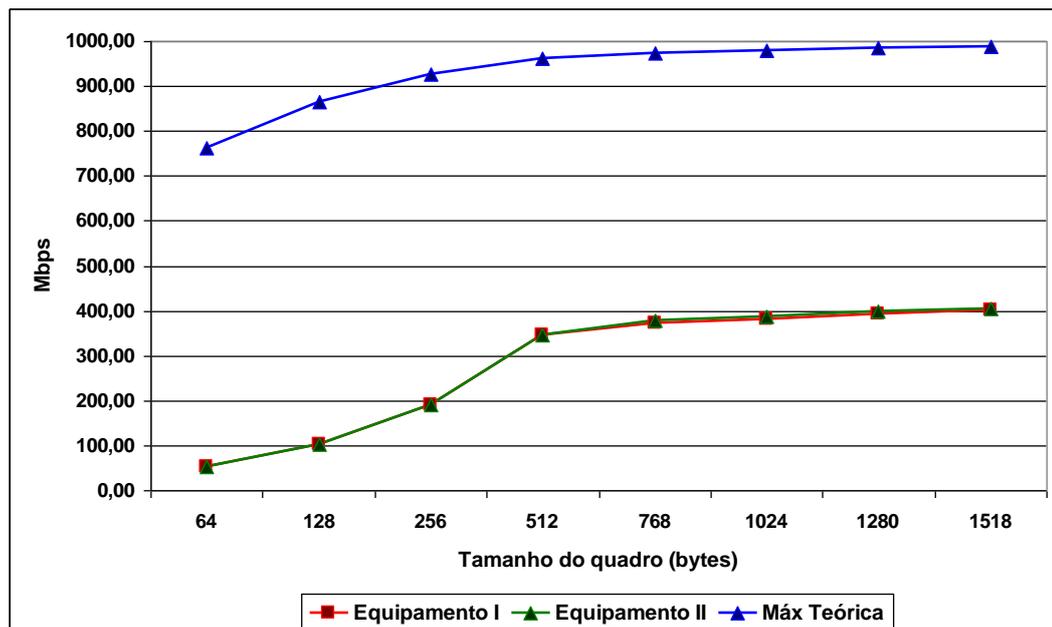


Figura 31 - Representação gráfica dos resultados em comparação com os limites teóricos, em taxa de bits e escala linear

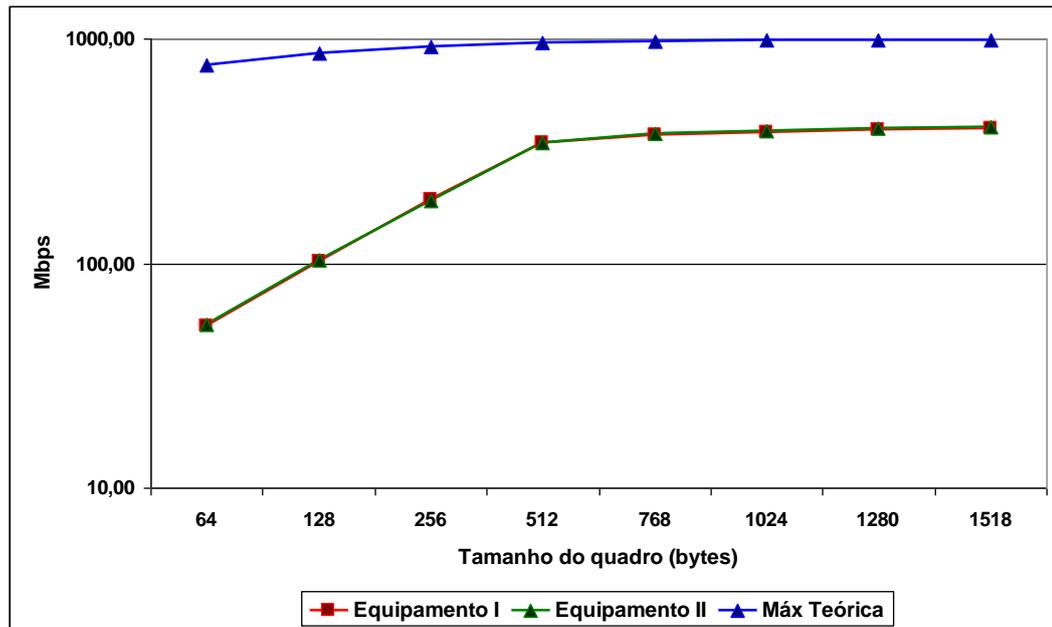


Figura 32 - Representação gráfica dos resultados em comparação com os limites teóricos, em taxa de bits e escala logarítmica

No desenvolvimento deste trabalho, o módulo receptor foi implementado conforme a primeira versão apresentada, onde o retardo é calculado a partir do número de quadros em trânsito. É importante mencionar que o retardo total, resultado deste teste, pode ser visto como o somatório de uma série de retardos menores entre o transmissor e o receptor.

$$R_t = R_f + R_b + R_p$$

Onde:

- R_f : retardo devido ao tempo de propagação do sinal nas fibras ópticas. Em taxas elevadas, acima de 1Gbps, os efeitos desse componente se tornam mais pronunciados;
- R_b : retardo devido à espera nas filas dos equipamentos;
- R_p : retardo de processamento nos equipamentos.

Na tabela a seguir, são apresentados os resultados de retardo de ambos os equipamentos, representados em microssegundos.

Tamanho do quadro (Bytes)	Retardo (μs) Equipamento I	Retardo (μs) Equipamento II	Dif % entre os equips II e I
64	397	355	-10,579
128	338	281	-16,864
256	323	236	-26,935
512	286	293	2,448
768	271	273	0,738
1024	260	338	30,000
1280	416	358	-13,942
1518	233	261	12,017

Tabela 5 - Resultado dos testes, representando o retardo médio medido

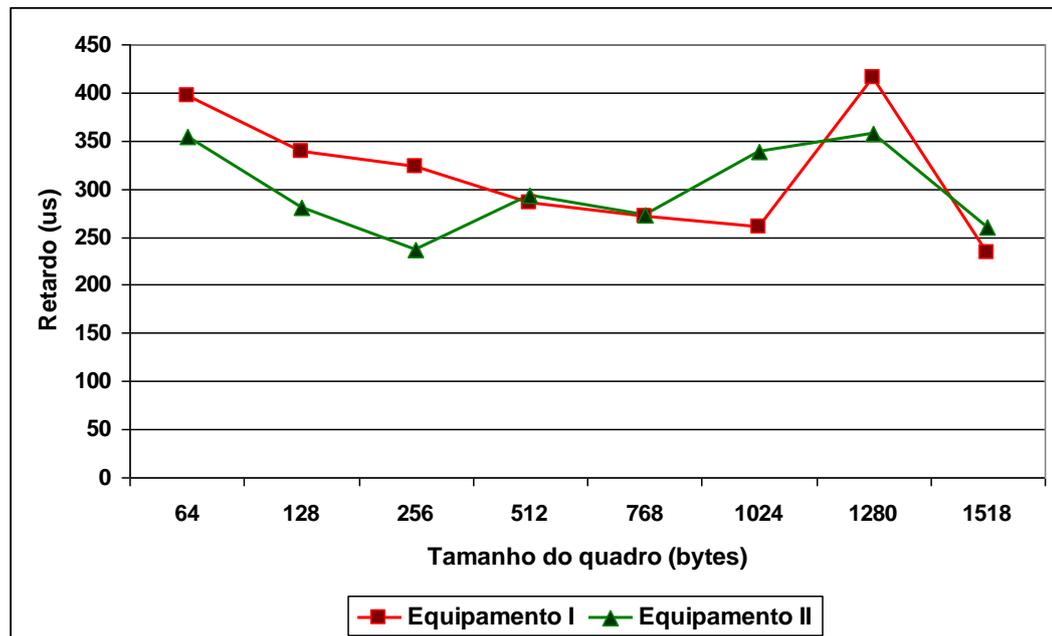


Figura 33 - Representação do resultado em forma gráfica, relacionando o retardo médio, medido por ambos os equipamentos, com o tamanho dos quadros

6.5.
Fotos da montagem para a realização dos testes



Figura 34 – Vista geral da montagem durante a realização dos testes

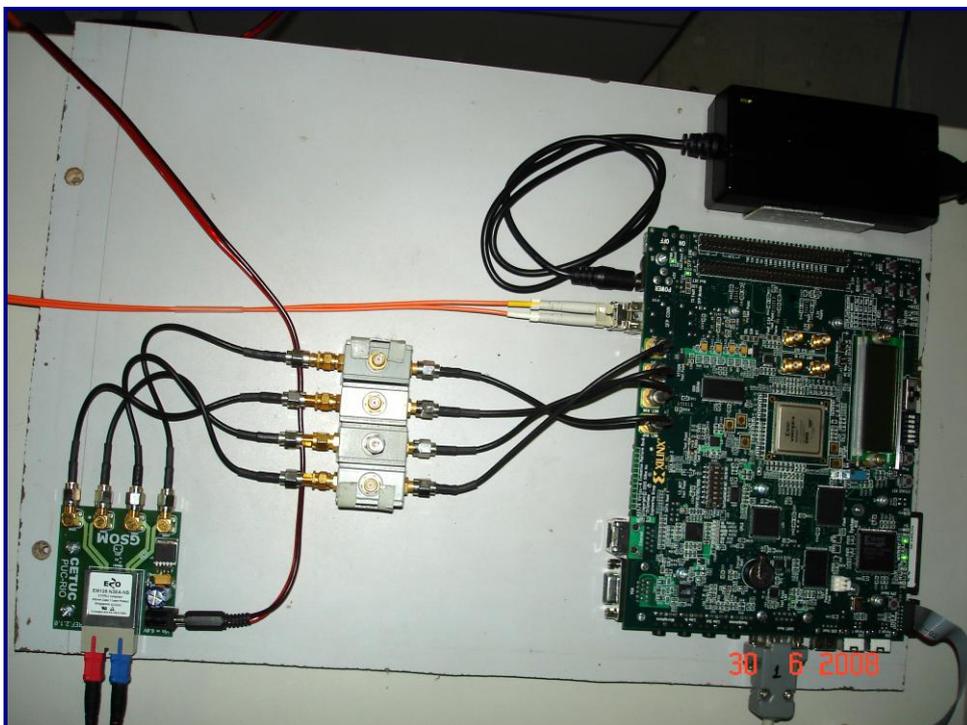


Figura 35 – Detalhe da placa principal e do módulo óptico externo

6.6. Comentários e Conclusões

Ao longo do presente capítulo foram apresentados os resultados dos testes com o módulo desenvolvido e o módulo de referência. Os resultados obtidos pelos dois equipamentos são muito próximos entre si. Considerando-se que o equipamento de referência foi produzido por um conhecido fabricante e com grande aceitação no mercado, pode-se reconhecer a validade dos testes realizados pelo equipamento proposto nessa dissertação.

Outra conclusão interessante diz respeito ao desempenho do elemento testado. Conforme poderia se esperar de um equipamento de uso geral, não específico para redes, sua capacidade é consideravelmente inferior aos limites teóricos da tecnologia de rede utilizada. Ainda assim, o equipamento proposto demonstrou a sua viabilidade de utilização em ambientes com baixo requisito de desempenho.

7

Comentários finais e conclusões

Ao longo do presente capítulo serão apresentados os resultados obtidos no desenvolvimento de um analisador de elementos de rede, capaz de avaliar a perda de quadros e o retardo associado a estes elementos. As principais vantagens do módulo desenvolvido serão destacadas e os principais problemas encontrados neste desenvolvimento serão avaliados.

7.1.

Observações finais

Inicialmente foram introduzidos alguns conceitos capazes de descrever o desempenho qualitativo e quantitativo de uma rede de telecomunicações. Em seguida, foi discutida a tecnologia FPGA destacando seu histórico, suas linguagens e seus principais blocos funcionais. A partir destes tópicos, um analisador de redes e elementos de rede foi proposto para desenvolvimento, destacando-se sua arquitetura básica. Em seguida foram descritas as etapas do desenvolvimento e da realização de um protótipo, onde foram incluídos detalhes da tecnologia adotada e as características da placa de desenvolvimento utilizada.

Finalmente, um conjunto de testes foi realizado envolvendo elementos de rede e trechos de fibra óptica. Os resultados obtidos foram comparados com outros obtidos através de um equipamento comercial já disponível no mercado.

7.2.

Principais resultados obtidos

Os principais resultados obtidos podem ser destacados através dos seguintes itens:

- O analisador realizado foi capaz de indicar corretamente a máxima capacidade de transmissão dos elementos testados na taxa de linha de 1 Gbps (1.25 Gbps);
- Pode ser utilizado qualquer tamanho de quadro permitido no padrão Ethernet, IEEE 802.3;
- A taxa de transmissão pode ser alterada de 100 Kbps até 1 Gbps, com variação discreta de 10 Kbps, mantendo-se a taxa de linha de 1 Gbps;
- O analisador realizado foi capaz de avaliar o retardo introduzido por uma conexão ou elemento de rede;
- Os resultados obtidos foram semelhantes aos fornecidos por um equipamento de referência, modelo Framescope Pro, da Agilent Technologies.

7.3.

Principais problemas e desvantagens

- O módulo desenvolvido utiliza ferramentas computacionais que avaliam o desempenho no nível das camadas 1 e 2 do modelo OSI[1]. Entretanto esta limitação permitiu uma dramática simplificação do referido módulo durante sua utilização, e poderá constituir um atrativo em inúmeras aplicações;
- O módulo desenvolvido opera apenas na taxa de linha de 1 Gbps (1,25 Gbps) por que utiliza uma versão proprietária das camadas inferiores do padrão Ethernet, com esta limitação ;
- O retardo medido utiliza um método simplificado e apresenta resultados aproximados;
- O módulo proposto não é capaz de medir a variação do retardo dos quadros (*Jitter*).

7.4. Comentários adicionais e trabalhos futuros

A quase totalidade dos analisadores estudados utiliza a norma RFC 2544[21] para avaliar as redes e os elementos de redes. Este procedimento adiciona complexidade e aumenta de forma significativa a duração dos testes. Através de contatos mantidos com algumas operadoras que possuem redes metropolitanas Ethernet na taxa de 1 Gbps e com alguns usuários corporativos destas redes, verificou-se que a utilização do módulo desenvolvido seria adequada e significativamente mais rápida, prática e simples para avaliar o desempenho destas redes.

O custo da placa de desenvolvimento utilizada é de US\$ 500, excluindo-se o custo dos *softwares* de desenvolvimento. Uma estimativa inicial de custo para a produção comercial do módulo desenvolvido é de R\$ 4.000.

Para trabalhos futuros, podem ser destacados:

- Implantação de um novo mecanismo de avaliação de retardo utilizando a segunda sistemática proposta no item 4.4;
- Adaptação das camadas físicas utilizadas, para operação em 10 Mbps e 100 Mbps;
- Desenvolvimento de um conjunto de elementos em *hardware* e *software* capazes de avaliar a variação estatística do retardo (*Jitter*) a partir de um elevado volume de dados extraídos durante os testes já ativados;
- Desenvolvimento de uma nova ferramenta computacional para ativar operação na taxa de 10 Gbps.

8 Referências

- 1 TANENBAUM, A. Computer Networks. Prentice-Hall, 3 ed, 1996/4 ed 2003. ISBN: 0-13-349945-6
- 2 IEEE std 802.3, 2000 Edition. (Incorporating IEEE Std 802.3, 1998 Edition, IEEE Std 802.3ac-1998, IEEE Std 802.3ab-1999, IEEE Std 802.3z-1998 and IEEE Std 802.3ad-2000)
- 3 DAVIDSON, J.; PETERS, J. Voice Over IP Fundamentals. Cisco Press, 2000, cap 8.
- 4 BATES, R. Broadband Telecommunications Handbook. Second Edition, McGraw-Hill, 2002, pp 503. ISBN 0-07-139851-1
- 5 THE INTERNET ENGINEERING TASK FORCE. Request For Comments: 1242, julho 1991
- 6 NAVABI, ZAINALABEDIN. Embedded Core Design with FPGAs, McGraw-Hill Professional, 2006
- 7 PEDRONI, V. Circuit Design with VHDL, The MIT Press, agosto 2004. ISBN 978-0262162241
- 8 BHASKER, J. Verilog HDL Synthesis, A Practical Primer, Star Galaxy Pub, outubro 1998. ISBN 978-0965039154
- 9 XILINX Inc, Local-Link Interface Specification, Publicação eletrônica, 2005
- 10 XILINX Inc, OPB Serial Peripheral Interface (SPI) – Product Specification, Publicação eletrônica, 2006
- 11 XILINX Inc, Virtex 4 family overview, publicação eletrônica, setembro 2004
- 12 MENDONÇA, A.; Zelenovsky, R. PC e Periféricos – Um guia completo de programação. Editora Ciência Moderna, 1996, pag 388
- 13 SIEWERT, S. SoC drawer: Function allocation and specification, IBM, Novembro 2005. Disponível em: http://www.ibm.com/developerworks/power/library/pa-soc2/index.html?S_TACT=105AGX16&S_CMP=EDU. Acesso em junho 2008
- 14 KREUGER, R. Digital Clock Management in Virtex-4 Devices, Xilinx, Inc, 2005. Disponível em http://www.xilinx.com/publications/xcellonline/xcell_52/xc_pdf/xc_v4dcm52.pdf. Acesso em junho 2008
- 15 XILINX Inc. Virtex-4 RocketIO Multi-Gigabit Transceiver User Guide, versão 3.2, publicação eletrônica, setembro 2006. Disponível em <http://www.slac.stanford.edu/exp/npa/xilinx/ug076.pdf>. Acesso em junho 2008
- 16 XILINX Inc. LogicCore - Processor Local Bus (PLB), Publicação eletrônica, julho 2003. Disponível em http://www.xilinx.com/support/documentation/ip_documentation/plb_v34.pdf. Acesso em junho 2008
- 17 XILINX INC, LogicCore - PLB to OPB Bridge (v1.01a), publicação eletrônica, julho 2005. Disponível em http://www.xilinx.com/support/documentation/ip_documentation/plb2opb_bridge.pdf. Acesso em junho 2008
- 18 SPIRENT COMMUNICATIONS. How to Test 10 Gigabit Ethernet Performance. White Paper, Maio 2005. Disponível em <http://www.spirent.com/documents/3731.pdf>. Acesso em junho 2008.

- 19 THE INTERNET ENGINEERING TASK FORCE. Request For Comments: 826, An Ethernet Address Resolution Protocol, novembro 1982.
- 20 MICROSOFT CORPORATION, MSDN Training: Programming with C#, fevereiro 2002
- 21 THE INTERNET ENGINEERING TASK FORCE. Request For Comments: 826, Benchmarking Methodology for Network Interconnect Devices, março 1999

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)