

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

TÍTULO DO TRABALHO

Projeto e Implementação em DSP de Sistema de
Comunicação Multiportadora Baseado no Padrão
G.fast

NOME DO AUTOR

Igor Mesquita de Almeida

DM:24/2013

UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
2013

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

NOME DO AUTOR

Igor Mesquita de Almeida

TÍTULO DO TRABALHO

Projeto e Implementação em DSP de Sistema de
Comunicação Multiportadora Baseado no Padrão
G.fast

DM:24/2013

UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
2013

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

NOME DO AUTOR

Igor Mesquita de Almeida

TÍTULO DO TRABALHO

**Projeto e Implementação em DSP de Sistema de
Comunicação Multiportadora Baseado no Padrão
G.fast**

Dissertação submetida à Banca Examinadora do Programa de Pós-graduação em Engenharia Elétrica da UFPA para a obtenção do Grau de Mestre em Engenharia Elétrica, ênfase em Telecomunicações.

UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
2013

Dados Internacionais de Catalogação-na-Publicação (CIP)
Sistema de Bibliotecas da UFPA

Almeida, Igor Mesquita de, 1988-
Projeto e Implementação em DSP de Sistema de
Comunicação Multiportadora Baseado no Padrão
G.fast / Igor Mesquita de Almeida. - 2013.

Orientador: Aldebaro Barreto da Rocha Klautau
Junior.

Dissertação (Mestrado) - Universidade Federal
do Pará, Instituto de Tecnologia, Programa de
Pós-Graduação em Engenharia Elétrica, Belém,
2013.

1. Processamento de sinais. 2. Sistema de
Comunicação em banda larga. I. Título.

CDD 23. ed. 621.3822

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Projeto e Implementação em DSP de Sistema de Comunicação Multiportadora Baseado no Padrão G.fast

AUTOR: Igor Mesquita de Almeida

DISSERTAÇÃO DE MESTRADO SUBMETIDA À AVALIAÇÃO DA BANCA EXAMINADORA APROVADA PELO COLEGIADO DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA, SENDO JULGADA ADEQUADA PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA ELÉTRICA NA ÁREA DE TELECOMUNICAÇÕES.

APROVADA EM 30/08/2013

BANCA EXAMINADORA:

.....
Prof. Dr. Aldebaro Barreto da Rocha Klautau Júnior (ORIENTADOR - UFPA)

.....
Prof. Dr. Evaldo Gonçalves Pelaes (MEMBRO - UFPA)

.....
Prof. Dr. Francisco Carlos Bentes Frey Müller (MEMBRO - UFPA)

.....
Prof. Dr. Johelden Campos Bezerra (MEMBRO - IFPA/IESAM)

VISTO:

.....
Prof. Dr. Carlos Renato Lisboa Francês

Vice-Coordenador do PPGEE/ITEC/UFPA

Agradecimentos

O trabalho cujo progresso esta dissertação se propõe a reportar é fruto de um esforço dedicado ao aprendizado, à experimentação e à partilha e construção coletiva do conhecimento. Como tal, seria egoísta e injusto não reconhecer o papel dos que contribuíram ou contribuem para o avanço dessa caminhada.

À família, avós, tios e primos, que me ajudaram a crescer em um ambiente acolhedor e alegre, obrigado pelo encorajamento e carinho, sempre presentes.

Mais especificamente, a meu pai, Alcy, e minha mãe, Judith, pela continuada dedicação e suporte ao longo dos anos; pelo amor que persevera a despeito da distância; pela genuína alegria demonstrada mesmo com as mais ínfimas vitórias e pela incondicional certeza nos sucessos dos quais eu mesmo duvidava, muito obrigado.

À minha irmã, Maísa, cujo próprio crescimento continua servindo de inspiração pessoal, agradeço pela companhia diária e pelo apoio, que certamente tornaram a vida de mestrando mais alegre.

Agradeço à Silvia, minha amada e companheira, cujo amor me fez capaz de ouvir e entender as estrelas.

Aos colegas de trabalho do LaPS e LASSE, obrigado por criarem um ambiente de colaboração tão agradável de se estar. Agradeço ao Ilan e Leonardo, em especial, por participarem da construção desse trabalho capitaneando seus respectivos projetos. É um prazer fazer parte dessa equipe.

Ao orientador de longa data, Aldebaro, muito obrigado pela inspiração e oportunidades concedidas.

Igor Mesquita de Almeida

Resumo

As redes de acesso usando cabos de cobre atualmente utilizam bandas de frequência até 30 MHz, especificada no padrão VDSL2. À medida que arquiteturas híbridas de fibra e cobre se tornam mais proeminentes na indústria e academia, torna-se possível utilizar cabos metálicos mais curtos (i.e. até 250 metros) conectando o último ponto de distribuição aos usuários, de modo que frequências mais altas podem ser exploradas para se alcançar taxas de transmissão de dados de 500 Mbps ou mais, como é o caso do padrão G.fast atualmente em desenvolvimento no ITU-T. Nesse trabalho, um simulador no domínio do tempo foi desenvolvido para avaliar a capacidade do sistema G.fast com diferentes tamanhos de extensão cíclica e diferentes topologias de rede especificadas pelo ITU-T. Os resultados das simulações mostram que sistemas G.fast são robustos a *bridged taps* e capazes de atingir altas taxas de dados para todas as topologias simuladas, provendo suporte à próxima geração de serviços de banda larga. Além disso, esse trabalho descreve o progresso da implementação de um protótipo de modem baseado no padrão G.fast em um ambiente híbrido de DSP *multicore* e FPGA utilizando kits de avaliação adquiridos pela UFPA. Arquiteturas, protocolos de comunicação e *benchmarks* são apresentados e avaliados para se chegar à conclusão de que tal protótipo é factível e fornece suporte flexível a várias linhas de pesquisa em banda larga da próxima geração.

PALAVRAS-CHAVE: G.fast; banda larga; FTTdp; simulação; implementação; DSP; FPGA.

Abstract

The evolving broadband access systems using copper networks are currently deployed in a frequency band that goes up to 30 MHz, as specified in VDSL2. As hybrid fiber-copper architectures become more important in the industry and academia, using shorter loop lengths (i.e. up to 250 meters) from the last distribution point to users enables adopting even higher frequencies to achieve very high data rates of 500 Mbps and beyond, as is the case with the G.fast standard under development by ITU-T. In this work, a time-domain simulator has been developed to evaluate G.fast system performance with different cyclic extension lengths and different reference loop topologies specified by ITU-T. The simulation results show that G.fast systems are robust to bridged taps and capable of providing very high data rates for all simulated loop topologies to support next generation ultra high speed broadband services. Furthermore, this work describes an ongoing effort to implement a G.fast-based modem prototype in a hybrid multicore DSP and FPGA environment using readily-available evaluation kits purchased by UFPA. Architecture plans, communication protocols and benchmarks are presented and evaluated to conclude that such a prototype is both feasible and flexible to support several avenues of research in next generation broadband.

KEYWORDS: G.fast; broadband; FTTP; simulation; implementation; DSP; FPGA.

Sumário

Lista de Figuras	iii
Lista de Tabelas	v
Lista de Abreviaturas	vi
1 INTRODUÇÃO	1
1.1 Tecnologias e arquiteturas DSL	3
1.2 Contribuições e organização do trabalho	5
2 BANDA LARGA DE QUARTA GERAÇÃO	7
2.1 Dificuldades típicas em sistemas DSL	8
2.2 O padrão G.fast	11
2.2.1 Modelos de referência	12
2.2.2 A camada física do padrão G.fast	16
2.2.2.1 Procedimento de inicialização	19
3 SIMULAÇÃO DA CAMADA FÍSICA DO SISTEMA G.FAST	21
3.1 Simulador de PHY G.fast no domínio do tempo	22
3.1.1 Etapas de simulação	22
3.1.2 Teoria e implementação	23
3.1.2.1 Aquisição de <i>timing</i> : detecção de preâmbulo	24
3.1.2.2 Equalização no domínio da frequência	24
3.1.2.3 Estimação de SNR	25
3.1.2.4 <i>Bitloading</i>	25
3.1.3 Verificação da implementação	27

3.2	Simulação de capacidade em cenários de referência	29
3.2.1	Abordagem	29
3.2.2	Topologias de referência do ITU-T simuladas	30
3.2.3	Configuração da simulação	30
3.2.4	Tamanho ótimo da extensão cíclica	32
3.2.5	Estudo <i>rate-reach</i>	34
4	PROTÓTIPO DE TRANSCEPTOR BASEADO EM G.FAST	36
4.1	Características do DSP	38
4.2	Arquitetura e desenvolvimento da aplicação	42
4.3	Testes de utilização do <i>interconnect</i> SRIO	44
5	CONSIDERAÇÕES FINAIS	46
	Bibliografia	51

Lista de Figuras

1.1	Conexões de acesso banda larga, Q4 2011. Fonte: Broadband Forum. . . .	2
1.2	Acesso à banda larga fixa no Brasil ao longo da última década. Fonte: Telebrasil.	2
1.3	Evolução da tecnologia DSL.	4
1.4	Exemplo de arquitetura de rede de acesso utilizando DSL.	4
1.5	A evolução da arquitetura da rede de acesso com as gerações de tecnologia DSL.	5
2.1	Exemplo de cenário de implantação em pequena cidade na Suécia.	8
2.2	Impacto do ruído impulsivo na transmissão DMT.	9
2.3	NEXT.	9
2.4	FEXT.	9
2.5	Um <i>bridged tap</i> em par trançado.	10
2.6	Emprego do prefixo cíclico em um símbolo DMT.	11
2.7	Coexistência entre xDSL e G.fast.	12
2.8	Modelo de referência para FTTdp.	13
2.9	Modelos de referência para DPU e transceptores G.fast. Fonte: <i>Draft</i> do padrão, julho de 2013.	14
2.10	Modelo de referência do protocolo do plano de dados. Fonte: <i>Draft</i> do padrão, julho de 2013.	15
2.11	<i>Framing</i> TDD adotado no G.fast.	17
2.12	Modelo de referência da subcamada PMD. Fonte: <i>Draft</i> do padrão, julho de 2013.	18
2.13	Modelo de referência da subcamada PMS-TC. Fonte: <i>Draft</i> do padrão, julho de 2013.	19

3.1	Diagrama de blocos do simulador.	23
3.2	Resposta ao impulso do cabo de 200 m obtido com medições.	27
3.3	Resultados para <i>bitloading</i> e taxa de erro de bits por tom.	28
3.4	Topologias do ITU-T simuladas.	30
3.5	Resposta ao impulso de cabo CAD55 com comprimento de 50 metros gerado com largura de banda de 100 MHz sem filtragem adicional versus canal filtrado usando largura de banda de 200 MHz. Nota: os coeficientes foram normalizados e deslocados para comparação visual.	31
3.6	Impacto da duração da extensão cíclica na capacidade nas topologias 1 e 4.	33
3.7	Impacto do aumento do comprimento do cabo na capacidade.	35
4.1	Visão geral da plataforma de prototipagem.	36
4.2	Diagrama de blocos do AFE.	37
4.3	Diagrama de blocos do protótipo.	37
4.4	Diagrama de blocos do TMS320C6670. Fonte: <i>Texas Instruments TMS320C6670 Data Manual</i>	38
4.5	Diagrama de blocos da placa TMDXEVM6670L. Fonte: <i>Texas Instruments TMDXEVM6670L Technical Reference Manual</i>	39
4.6	Diagrama de blocos do <i>Multicore Navigator</i> . Fonte: <i>Texas Instruments Multicore Navigator User Guide</i>	39
4.7	Enfileiramento de descritores no <i>Multicore Navigator</i> . Fonte: <i>Texas Instruments Multicore Navigator User Guide</i>	40
4.8	Transmissão e recepção de pacotes no QMSS.	41
4.9	Arquitetura da aplicação para o protótipo.	42
4.10	Resultados de <i>benchmark</i> para transações NREAD e NWRITE_R.	45

Lista de Tabelas

2.1	Comparação entre parâmetros do G.fast e VDSL2.	17
3.1	Parâmetros de simulação.	31
3.2	Tamanhos ótimos para extensão cíclica (em microssegundos).	32

Lista de Abreviaturas

ADC *Analog-to-Digital Converter*

ADSL *Asymmetric Digital Subscriber Line*

AFE *Analog Front-end*

AMC *Advanced Mezzanine Card*

AN *Access Node*

ANSI *American National Standards Institute*

AWG *American Wire Gauge*

BER *Bit Error Rate*

BSD *Berkeley Software Distribution*

CAB *Street cabinet*

CO *Central Office*

DAC *Digital-to-Analog Converter*

DFT *Discrete Fourier Transform*

DMA *Direct Memory Access*

DMT *Discrete multi-tone modulation*

DP *Distribution Point*

DPU *Distribution Point Unit*

DQM *DSL Quality Management*

DRA *Dynamic Resource Allocation*

DRR *Dynamic Resource Reports*

DSL *Digital Subscriber Line*

DSLAM *Digital Subscriber Line Access Multiplexer*

DSP *Digital Signal Processor*

DTU *Data Transfer Unit*

E-PON *Ethernet Passive Optical Network*

FDD *Frequency Division Duplexing*

FEC *Forward Error Correction*

FEQ *Frequency-domain equalizer*

FEXT *Far-end Crosstalk*

FFT *Fast Fourier Transform*

FFTC *FFT Coprocessor*

FM *Frequency Modulation*

FPGA *Field-Programmable Gate Array*

FTTdp *Fiber to the last distribution point*

FTTH *Fiber to the Home*

FTTX *Fiber to the X*

FTU *G.fast Transceiver Unit*

G-PON *Gigabit Passive Optical Network*

HDSL *High bit rate Digital Subscriber Line*

HON *Higher Order Node*

IBGE *Instituto Brasileiro de Geografia e Estatística*

IDFT *Inverse Discrete Fourier Transform*

IFFT *Inverse Fast Fourier Transform*

IP *Internet Protocol*

ISI *Intersymbol interference*

ITU *International Telecommunication Union*

ITU-T *International Telecommunication Union - Telecommunication Standardization Sector*

LFSR *Linear Feedback Shift Register*

LNA *Low-noise Amplifier*

LVDS *Low-voltage Differential Signalling*

ME *Management Entity*

MER *Modulation Error Ratio*

NETCP *Network Coprocessor*

NEXT *Near-end Crosstalk*

NMS *Network Management System*

NLMS *normalized least mean squares*

NT *Network Termination*

ODN *Optical Distribution Network*

PCE *Power Control Entity*

PGA *Programmable Gain Amplifier*

PKTDMA *Packet DMA*

PMD *Physical Medium Dependent*

PMS-TC *Physical Media Specific Transmission Convergence*

PON *Passive Optical Network*

PSD *Power Spectral Density*

QAM *Quadrature Amplitude Modulation*

QMSS *Queue Manager Subsystem*

RS *Reed-Solomon*

RTOS *Real-time operating system*

SDR *Software-defined radio*

SLA *Service Level Agreement*

SNR *Signal-to-noise ratio*

SoC *System-on-Chip*

SRIO *Serial RapidIO*

S-TDD *Synchronous Time Division Duplexing*

TCE *Timing Control Entity*

TCM *Trellis Coded Modulation*

TDD *Time Division Duplexing*

TP *Transport Protocol*

TPS-TC *Transport Protocol Specific Transmission Convergence*

VCE *Vectoring Control Entity*

VDSL *Very High bit rate Digital Subscriber Line*

WLAN *Wireless Local Area Network*

Capítulo 1

INTRODUÇÃO

Há mais de um século, a infraestrutura da rede telefônica global vem sendo construída, com um investimento acumulado de mais de um trilhão de dólares [1]. Ao longo desse tempo, os serviços oferecidos evoluíram em todos os aspectos: taxa de transmissão, disponibilidade e uso do espectro sendo os mais proeminentes.

Essa crescente demanda por conectividade tornou o acesso de banda larga uma infraestrutura básica para a sociedade conectada, de modo que em todos os países, essas redes se tornaram tão importantes quanto os meios de transporte, eletricidade ou sistema de abastecimento de água, por exemplo. Pode-se dizer, inclusive, que redes de banda larga fornecem também a fundação para outras infraestruturas de comunicação.

Hoje, essas redes transportam dados e proveem acesso a serviços e oportunidades a pessoas em todo o mundo. Segundo a União Internacional de Telecomunicações, em 2010 a quantidade de informação digital trafegada ultrapassou *um zettabyte* (10^{21} bytes) pela primeira vez. A *rede elétrica inteligente*, diagnósticos médicos à distância e aplicações de educação à distância [2, 3] são exemplos do uso da tecnologia para o desenvolvimento [4] e melhoria da qualidade de vida [5, 6].

Embora outros meios (fibra óptica, acesso *wireless*, *cable modem*, *power line communications*, etc.) possam ser usados para prover banda larga a assinantes, nenhum deles é tão ubíquo ou tem a mesma maturidade de desenvolvimento quanto a rede telefônica. Em 2006, o número de linhas telefônicas de par trançado instaladas era da ordem de centenas de milhões [7], atingindo meio bilhão de assinantes em todo o mundo ao fim de 2010 [5]. A Figura 1.1 mostra uma comparação dos principais meios de acesso banda larga fixa, indicando que a tecnologia mais difundida é a DSL (do inglês, *Digital Subscriber Line*) [8].

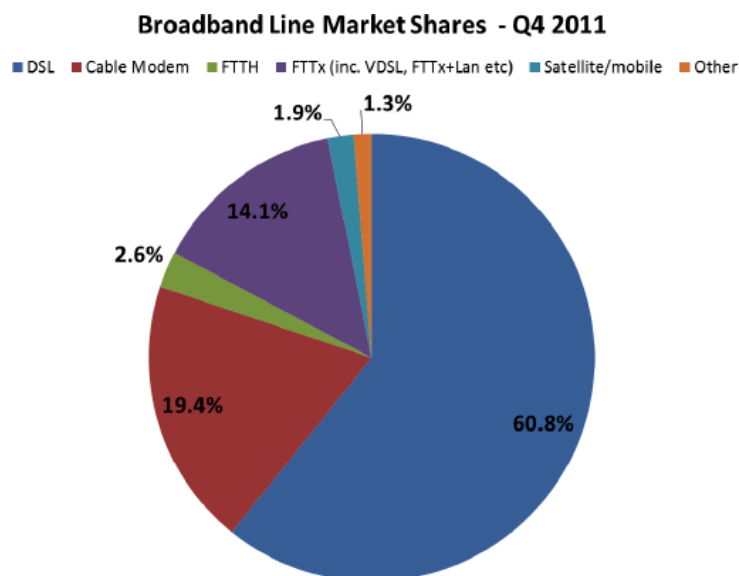


Figura 1.1: Conexões de acesso banda larga, Q4 2011. Fonte: Broadband Forum.

No Brasil, estima-se o número de assinantes de banda larga fixa (i.e., acesso a mais de 256 kbps) em mais de 21 milhões, número que vem crescendo como mostra a Figura 1.2, onde é possível ver que mais de 10 milhões de usuários assinam apenas a tecnologia ADSL [9]. No mesmo documento, a taxa de transmissão média da banda larga fixa no Brasil é reportada em 7,3 Mbps, embora a cobertura seja bastante desigual: em 2009, segundo o IBGE, 71% das casas do Distrito Federal possuíam provisão do serviço, enquanto no Pará esse número era de apenas 10,8% [4].

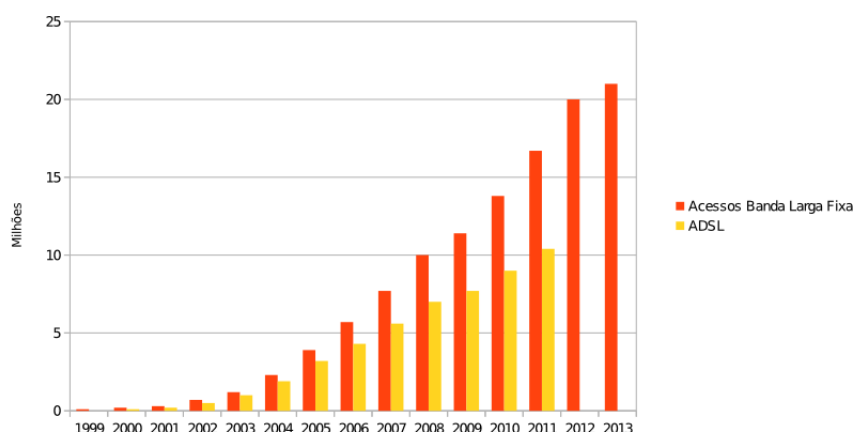


Figura 1.2: Acesso à banda larga fixa no Brasil ao longo da última década. Fonte: Telebrasil.

1.1 TECNOLOGIAS E ARQUITETURAS DSL

Atualmente existem inúmeras variantes da tecnologia DSL, o que reflete as necessidades diferenciadas de taxa de dados dos assinantes, como usuários domésticos, empresas de tamanho pequeno ou médio, escolas, etc. Em relação aos serviços exclusivamente assimétricos, as principais tecnologias DSL são:

- ADSL (*Asymmetric DSL*) [10];
- ADSL2 (ADSL versão 2) [11]; e
- ADSL2+ (ADSL2 com banda estendida) [12].

O ADSL suporta um raio de cobertura de até 5 km e taxas de transmissão de até 8 Mbps na direção *downstream* (que corresponde à transmissão em direção ao assinante) e até 1 Mbps na direção *upstream* (transmissão do assinante para a rede). Seu uso é recomendado para usuários domésticos e empresas de pequeno porte, que possuem uma demanda maior de *download* do que de *upload*. O ADSL2, por sua vez, melhorou o desempenho e interoperabilidade do ADSL, fornecendo suporte para novas aplicações e serviços, podendo alcançar taxas de até 12 Mbps na direção *downstream*, dependendo do tamanho do enlace e outros fatores. Por fim, o ADSL2+ melhorou ainda mais o ADSL2, aumentando sua largura de banda de *downstream* de 1,1 MHz para 2,2 MHz, o que se reflete em taxas de transmissão de até 25 Mbps nessa direção [13].

Em relação a serviços empresariais, as principais tecnologias DSL são:

- VDSL (*Very high speed DSL*) [14]; e
- VDSL2 (*VDSL versão 2*) [15].

O VDSL pode ser configurado tanto para serviços simétricos quanto assimétricos, podendo alcançar taxas de dados de até 52 Mbps na direção *downstream*. Porém, o raio de alcance dessa tecnologia fica em torno de 1,2 km. O VDSL é particularmente útil para suprir serviços com alta vazão de dados como hotéis e prédios comerciais, sem a necessidade de uma nova infraestrutura além da rede telefônica preexistente. Além disso, a tecnologia VDSL também é ideal para campus de universidades e parques empresariais, onde as distâncias para os roteadores de borda são pequenas.

Por fim, surgiu o VDSL2, que é atualmente a tecnologia mais moderna de DSL em operação, podendo alcançar taxas de até 100 Mbps. Quando surgiu, seu objetivo era

garantir o bom desempenho para linhas maiores que as suportadas pelo VDSL, como uma evolução da tecnologia ADSL2+, e também para linhas menores, como uma evolução do VDSL. Enquanto que o raio de alcance do VDSL gira em torno de 1,2 km, o alcance do VDSL2 pode ser estendido a até 2,4 km [13]. Nos últimos anos, VDSL2 vem sendo implantado no sul e sudeste do Brasil, com pacotes residenciais de taxas *downstream* variando entre 5 e 30 Mbps, podendo chegar a 100 Mbps no sul do país.

A Figura 1.3 mostra a evolução das principais tecnologias DSL, em termos de taxas de bit máximas suportadas. Para comparação, foi incluído no gráfico o HDSL (*High bit rate DSL*), uma das primeiras versões de sistema DSL que surgiu no mercado.

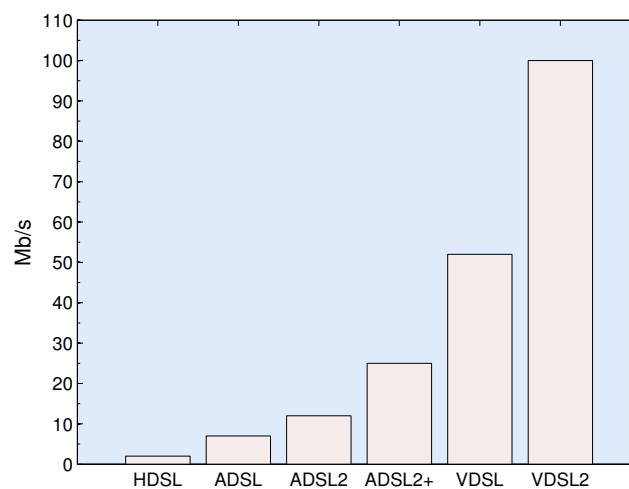


Figura 1.3: Evolução da tecnologia DSL.

Na Figura 1.4, um exemplo comum de arquitetura de rede de acesso é mostrado, em que um usuário se conecta à rede central diretamente através de um enlace DSL. Esse enlace é composto pelo *modem* do assinante (transceptor do usuário), o meio de transporte (cabo telefônico) e um *Digital Subscriber Line Access Multiplexer* (DSLAM), que faz o papel do transceptor da operadora e é capaz de se comunicar e gerenciar várias linhas simultaneamente.

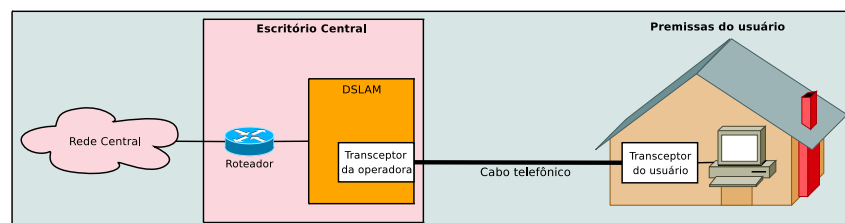


Figura 1.4: Exemplo de arquitetura de rede de acesso utilizando DSL.

Ao longo dos anos, a arquitetura da rede de acesso baseada em DSL vem mudando, e com o aumento da taxa de transmissão veio o uso de fibra óptica em trechos do enlace. Essa arquitetura híbrida, chamada genericamente de *Fiber to the X* (FTTX), é necessária devido à limitação dos cabos metálicos em fornecer taxas mais altas a longas distâncias. A implantação de fibra óptica na rede de acesso, entretanto, tem sido gradual, custosa e demorada [16]. Divide-se, então, as tecnologias DSL em gerações [17], como mostra a Figura 1.5.

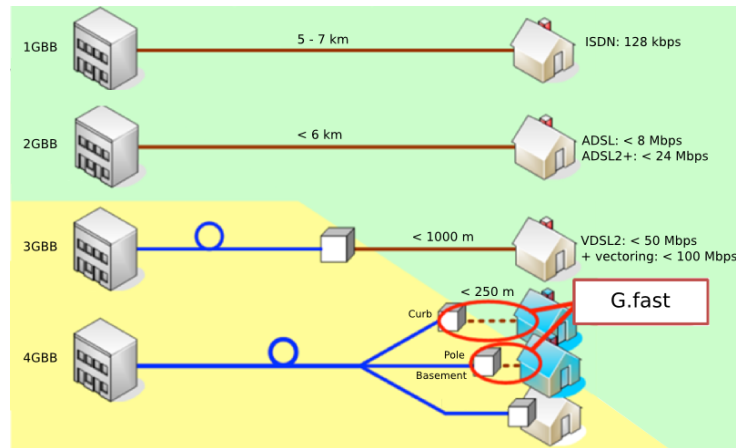


Figura 1.5: A evolução da arquitetura da rede de acesso com as gerações de tecnologia DSL.

Esse trabalho, portanto, acompanha o desenvolvimento da quarta geração de banda larga, representado pelo ainda incompleto padrão G.fast [18]. Projetado para prover serviços a taxas de transmissão próximas às de conexões usando redes completamente ópticas, o padrão se posiciona para reutilizar as últimas centenas de metros de cobre entre o usuário e o ponto de distribuição, diminuindo o gasto imediato requerido para suprir a sempre crescente demanda por capacidade.

1.2 CONTRIBUIÇÕES E ORGANIZAÇÃO DO TRABALHO

Os capítulos desta dissertação estão organizados da seguinte maneira:

- O Capítulo 2 apresenta o projeto de banda larga de quarta geração (4GBB), as dificuldades típicas enfrentadas por sistemas DSL e, finalmente, uma introdução ao padrão G.fast;
- O Capítulo 3 apresenta os primeiros esforços de desenvolvimento e simulação do sistema, visando avaliar sua capacidade e o impacto de alguns parâmetros em sua

performance: uma primeira contribuição, publicada em [19], mostrando a factibilidade de tal sistema em cenários realísticos definidos pela União Internacional de Telecomunicações - Setor de Normatização das Telecomunicações (ITU-T);

- O Capítulo 4 apresenta o desenvolvimento de um protótipo desse sistema utilizando um DSP *multicore* e um FPGA como módulos digitais conectados a uma interface analógica: uma segunda contribuição desta dissertação, apresentando a arquitetura de um modem baseado no padrão G.fast. Ao fim do capítulo, descreve-se os testes com o protótipo e os resultados obtidos com o periférico utilizado para comunicação com o FPGA;
- O Capítulo 5 apresenta as considerações finais, os próximos sistemas a ser implementados na aplicação principal e, por último, ideias para trabalhos futuros.

Capítulo 2

BANDA LARGA DE QUARTA GERAÇÃO

Operadores que usam a rede telefônica para oferecer serviços de banda larga são incentivados a aumentar as taxas de transmissão devido à forte competição e a planos governamentais. Ao gradualmente implantar fibra óptica em suas redes, os *access nodes* (ANs) são levados para cada vez mais perto dos usuários, portanto reduzindo o tamanho do cabo de cobre para menos de um quilômetro e aumentando sua capacidade. Usando a tecnologia VDSL2 com *vectoring* nesses canais, é possível alcançar taxas agregadas de 50 a 200 Mbps [16].

Entretanto, uma série de fatores aumentam a demanda por maiores taxas de transmissão, como o aumento do número de terminais conectados, do número de serviços por aparelho, o uso de videochamadas e conferências, *download* de jogos e aplicações multi-jogador [20]. De fato, a implantação de redes ópticas passivas (PON, do inglês *passive optical networks*) são uma abordagem adequada, em que a infraestrutura e a banda são compartilhadas entre os usuários de uma mesma área residencial.

Essa abordagem, chamada *Fiber to the home* (FTTH), é limitada apenas pelo custo da implantação. Em [17], o cenário de rede de acesso mostrado na Figura 2.1 é avaliado. Nesse cenário, um cabo primário conecta o escritório central (CO) a vários *street cabinets* (CABs) e, daí, progressivas ramificações permitem que se alcance as premissas do usuário.

Substituir todo o cobre por fibra a partir do ponto de distribuição no *cabinet* (na figura, *DP in CAB*) até o último DP implica cavar aproximadamente 5 km ($500 \text{ m} \times 10 \text{ bundles}$), e desse ponto para cada casa são adicionais 30 km por *cabinet* ($100 \text{ m} \times 30 \times 10$). Considerando o custo médio da operação em 105 mil euros por quilômetro,

calcula-se o investimento imediato em FTTH sendo um total de 3,67 milhões de euros.

A segunda abordagem, chamada *Fiber to the last distribution point* (FTTdp), substitui apenas os 5 km iniciais do cenário-exemplo, mantendo a conexão metálica até as premissas do usuário. A 525 mil euros, 14% do custo total, o investimento permite que os cabos de cobre, mais curtos e portanto com menor atenuação, possam ser explorados também em frequências mais altas, o que aumenta sua capacidade. A banda larga de quarta geração (4GBB), então, fornece um caminho de atualização mais gradual para a última milha.

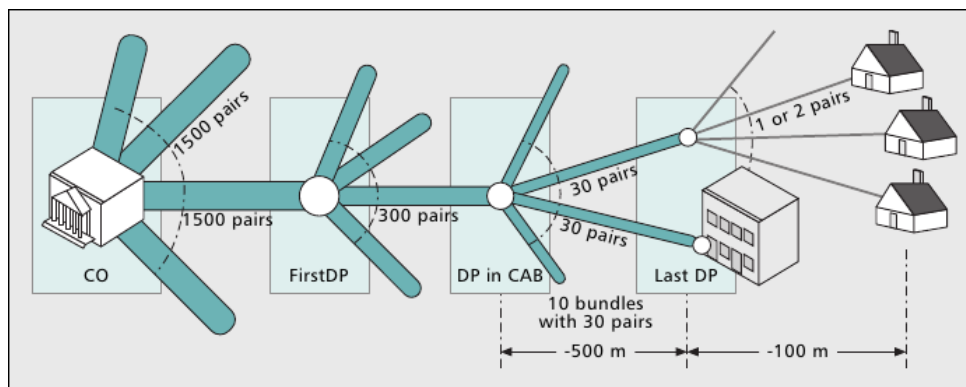


Figura 2.1: Exemplo de cenário de implantação em pequena cidade na Suécia.

2.1 DIFICULDADES TÍPICAS EM SISTEMAS DSL

Por ter sido construída inicialmente para transportar sinais de voz, a rede telefônica não foi projetada para combater alguns distúrbios chamados *limitantes de qualidade* ou *limitantes de capacidade* nas aplicações modernas de transmissão de dados [21].

Várias fontes de interferências e ruídos são, hoje, comuns em sistemas DSL. Alguns deles são geralmente intermitentes, geograficamente variados e de ocorrência difícil de ser prevista, como, por exemplo, ruídos impulsivos como o mostrado na Figura 2.2, tipicamente originados de descargas elétricas, cercas eletrificadas, acionamento de motores, lâmpadas, etc., de modo que sistemas DSL devem conter algum mecanismo de *impulse noise protection* [22] para evitar que limitem consideravelmente a qualidade do sinal.

Ruídos limitantes de capacidade, por outro lado, como o ruído térmico e o *crosstalk*, possuem estatísticas que tendem a variar menos com o tempo e, por isso, são previstos com mais facilidade, permitindo que seu impacto possa ser mitigado.

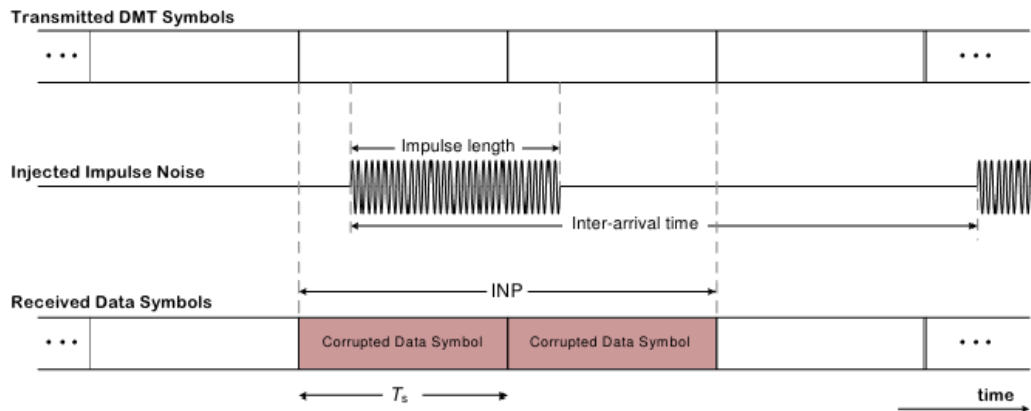


Figura 2.2: Impacto do ruído impulsivo na transmissão DMT.

Crosstalk

O *crosstalk* é o acoplamento magnético que ocorre entre pares telefônicos adjacentes e é considerado o maior limitante de capacidade dos sistemas DSL, em termos de taxa de transmissão ou raio de alcance [13].

O *crosstalk* é dividido de acordo com a extremidade da linha em que está localizado o receptor da interferência, existindo portanto dois tipos, denominados de *near-end crosstalk* (NEXT) e *far-end crosstalk* (FEXT). O NEXT é a interferência proveniente do acoplamento do sinal no receptor localizado na mesma extremidade do cabo que o transmissor (*near-end*), afetando sua recepção como mostrado na Figura 2.3. O FEXT, por outro lado, afeta a recepção de um usuário na outra extremidade da linha (*far-end*), como mostrado na Figura 2.4.

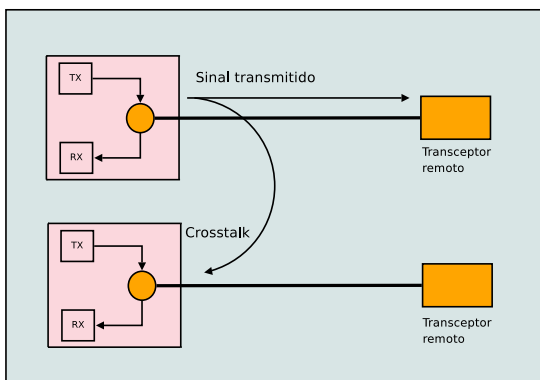


Figura 2.3: NEXT.

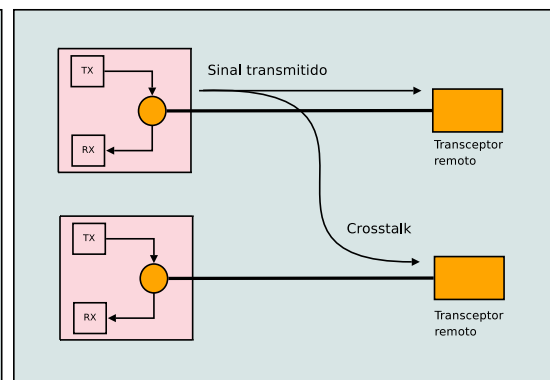


Figura 2.4: FEXT.

A intensidade da interferência de NEXT é bem maior que a de FEXT, visto que esse último é bastante atenuado antes de atingir o receptor da extremidade oposta do

cabo. As tecnologias atuais de DSL evitam o NEXT utilizando técnicas de duplexação por divisão em frequência (FDD). Por esse motivo, costuma-se desprezar os efeitos do NEXT em simulações de sistemas DSL. Sendo assim, o FEXT é tido como a principal fonte de interferência de *crosstalk* e, para combatê-lo, a técnica conhecida como *vectoring* [23] é, atualmente, a mais utilizada, padronizada pela *Recommendation ITU-T G.993.5* [24].

O objetivo do *vectoring* é aprender e manter a matriz de *crosstalk* em frequências determinadas nas bandas de operação e aplicar o processamento de sinais necessário para cancelar a interferência gerada pelos outros transceptores compartilhando o mesmo *binder*, que, agrupados, são chamados de *vectored group*. Isso é possível adicionando-se um elemento sincronizador chamado *vectoring control entity* que configura precodificadores presentes nos transceptores, de modo que o *crosstalk* no grupo seja cancelado.

Uma série de questões operacionais e arquiteturais devem ser atendidas para que os benefícios do *vectoring* sejam aproveitados. Entre eles, o aumento na complexidade do gerenciamento do cabo e do *binder*; a seleção das linhas que mais se beneficiarão do *vectoring*; e garantir que ruído não relacionado a *crosstalk* seja mitigado por técnicas de *DSL Quality Management (DQM)* [8], uma vez não é possível cancelar o *crosstalk* vindo de linhas fora do grupo.

Bridged taps

É comum em redes telefônicas que um par trançado esteja conectado a outro em algum ponto ao longo de seu comprimento, e geralmente essa bifurcação é mantida em circuito aberto quando deixa de ser usada. Isso é conhecido como *bridged tap* e é ilustrado na Figura 2.5.

Bridged taps podem, inclusive, existir em multiplicidade na mesma linha e, geralmente, são criadas quando há a terminação de serviço para algum assinante e, concomitantemente, há a necessidade de se prover serviço a um novo cliente na vizinhança [7]. Embora seu impacto na banda de telefonia seja negligível, sua existência causa reflexões do sinal que, por sua vez, alteram a função de transferência do canal, inserindo nulos espectrais e distorção de fase na banda de operação do sistema DSL.

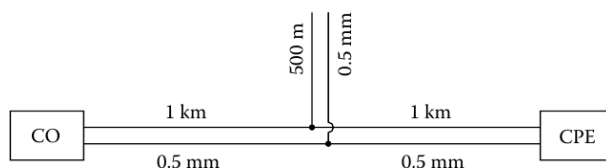


Figura 2.5: Um *bridged tap* em par trançado.

Redundância versus capacidade

Para combater o impacto de *bridged taps* e, mais geralmente, da dispersão do canal, emprega-se o prefixo cíclico, que nada mais é que a cópia das últimas amostras do símbolo DMT, enviadas antes desse, como mostra a Figura 2.6. Se a dispersão do canal não for maior que a duração desse intervalo de guarda, ao se descartar o prefixo cíclico na recepção não haverá interferência intersimbólica (ISI) e, assim, o sinal pode ser recuperado após a equalização no domínio da frequência.

De fato, para evitar a ISI, é suficiente que o intervalo de guarda seja nulo. Porém, outra importante contribuição do prefixo cíclico é manter a ortogonalidade entre as sub-portadoras. Isso é possível pois, com a introdução do prefixo cíclico, a convolução realizada pelo canal, assumido linear e invariante no tempo, passa a simular uma convolução circular.

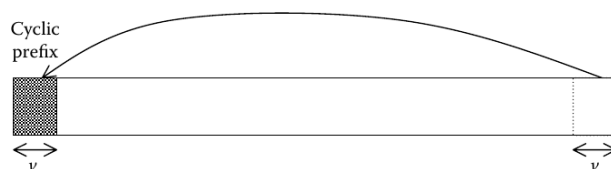


Figura 2.6: Emprego do prefixo cíclico em um símbolo DMT.

Claramente, a adição do prefixo cíclico diminui a taxa de símbolos e portanto a capacidade do sistema, uma vez que parte do tempo do símbolo é dedicado à transmissão de informação redundante. Há, então, um compromisso a ser feito entre a proteção contra ISI e a maior taxa de transmissão advinda da diminuição do tempo do símbolo [25].

2.2 O PADRÃO G.FAST

Iniciado em fevereiro de 2011, o novo padrão do ITU prevê taxas de transmissão de até 1 Gbps usando os cabos de cobre existentes. Em julho de 2013, o primeiro estágio de aprovação foi concedido à *Recommendation ITU-T G.9700*, que especifica métodos para eliminar o risco de interferência entre equipamentos G.fast e serviços de transmissão em massa como rádio FM [18]. Além de acesso residencial comum, o padrão é também apropriado para *backhaul* de *small cells* e pontos de acesso *Wi-Fi* (WLAN IEEE 802.11) [26].

Para facilitar a adoção, algumas características são mandatórias aos provedores de serviço:

- Baixa potência, baixa complexidade, baixo custo;
- Alimentação *reversa*, isso é, a alimentação do equipamento no ponto de distribuição é fornecida pelo *gateway* residencial do assinante;
- Instalação fácil realizada pelo próprio assinante;
- Controle da taxa de assimetria *downstream/upstream*: com configurações mandatórias 90%/10% e 50%/50% e opcionais de 50%/50% a 10%/90%;

Além dessas características mandatórias, outro importante aspecto é a coexistência com tecnologias xDSL. A Figura 2.7 mostra como a frequência inicial de operação pode ser alterada para evitar interferência entre os serviços. Também mostrada na figura é a banda de operação de 100 MHz, em uma primeira versão, e 200 MHz em futuras adições ao padrão.

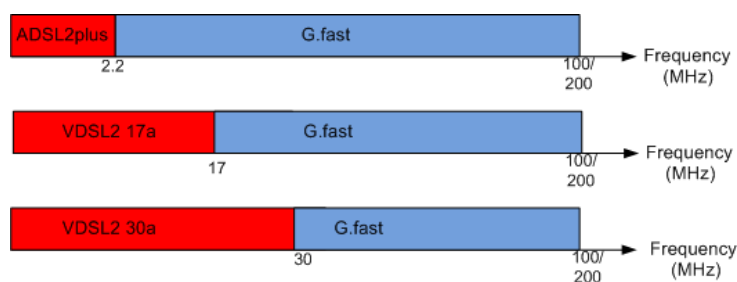


Figura 2.7: Coexistência entre xDSL e G.fast.

2.2.1 Modelos de referência

O atual modelo de referência para implantação de FTThp é mostrado na Figura 2.8. Partindo da esquerda, os blocos HON (*Higher Order Node*) e ODN (*Optical Distribution Network*) indicam a conexão do enlace G.fast com a rede óptica, que agrega o tráfego de todos os *distribution point units* (DPUs) [27], sendo um DPU o correspondente ao bloco tracejado na figura. Tal agregação passa pela camada física representada pelo bloco PHY e pode, por exemplo, consistir de rede G-PON, E-PON, rede gigabit Ethernet ponto-a-ponto, entre outras, ficando a critério da empresa implantando o sistema. Finalizando a representação da conexão com rede óptica, o bloco L2+ dentro do DPU representa a Camada 2 e camadas superiores implementadas no DPU.

Os blocos mencionados no parágrafo anterior estão fora do escopo do padrão G.fast e, portanto, não são especificados além do necessário para completar o entendimento do

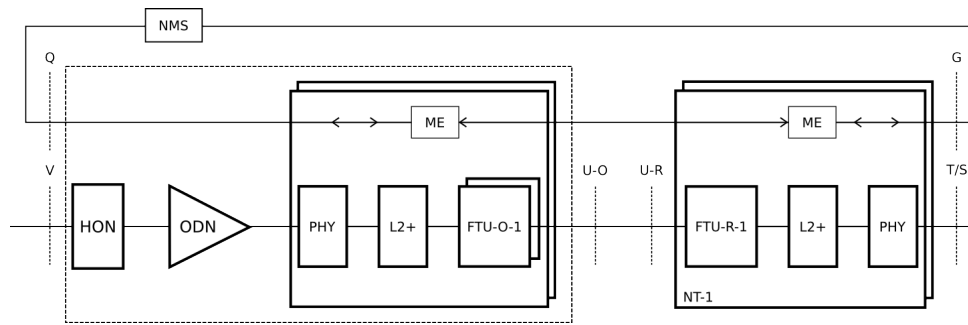


Figura 2.8: Modelo de referência para FTTdp.

fluxo de dados. De maior interesse nesse trabalho são os transceptores FTU-O e FTU-R, cuja interface é objeto da especificação descrita nesse capítulo e apresentada com mais detalhes a seguir.

Um DPU pode conter um ou mais transceptores FTU-O, cada um deles conectado pelo par trançado a um transceptor FTU-R dentro do *network termination point* (NT), localizado nas premissas do usuário. Da mesma forma que o DPU, o NT possui o bloco L2+, com funcionalidade similar; e o bloco PHY, representando a camada física da conexão com a rede interna do usuário, tipicamente rede IP em gigabit Ethernet ou *Wi-Fi*.

Finalmente, a Figura 2.8 mostra a existência de *management entities* (MEs) no DPU e no NT, responsáveis pela comunicação com o *network management system* (NMS), todos definidos na *Recommendation ITU-T G.997.1* [28].

O modelo de referência dos transceptores é mostrado na Figura 2.9. Além da camada física G.fast propriamente dita, representada pelos blocos FTU-O-n e FTU-R, alguns blocos extras mostram as características do sistema, de certa forma impostos pela arquitetura da rede e pelas estratégias escolhidas para facilitar a adoção, e são descritos nos parágrafos a seguir.

Começando pelo DPU na Figura 2.9a, os transceptores FTU-O-n comunicam-se pela interface U com o NT, primariamente, trocando as informações representadas por STREAMds-n, no papel de *downstream*, e STREAMus-n, no papel de *upstream*. Controle de fluxo pode também ser usado (FCTLds-n). Além disso, os seguintes blocos estão presentes:

- *Timing Control Entity* (TCE);
- *Vectoring Control Entity* (VCE);
- *Dynamic Resource Allocation* (DRA);

STREAM-BC-n (vinda do FTU-R) para determinar os coeficientes do precodificador.

O DRA, por sua vez, coordena as oportunidades de transmissão para *downstream* e *upstream* no grupo. Essa alocação pode ser estática (i.e., definida previamente pelo ME) ou dinâmica (i.e., reconfigurável, sujeita aos limites impostos pelo ME e pelo PCE descrito abaixo). Os *Dynamic Resource Reports* (DRR) para cada linha são transmitidos pelos NT e, combinados com a informação de gerenciamento, são usados para configurar os limites de alocação de recursos para *downstream* e *upstream*, definidos por exemplo em *service level agreements* (SLAs), taxa de assimetria, etc.

Finalmente, o *power control entity* (PCE) pode também impor limites na alocação de recursos, usando informações como o tráfego requerido por linha, relatórios de temperatura, potência e qualidade das fontes de alimentação no DPU, consumo de potência por usuário, etc.

O modelo de referência para o NT é consideravelmente mais simples, como mostra a Figura 2.9b, uma vez que não há mais de um transceptor na mesma terminação. Caso necessário, controle de fluxo pode ser usado para a interface STREAMus e, diferente da arquitetura do DPU, o controle de temporização para S-TDD é realizado inteiramente no FTU-R, deixando o TCE apenas com a função de repassar as referências temporais às camadas superiores.

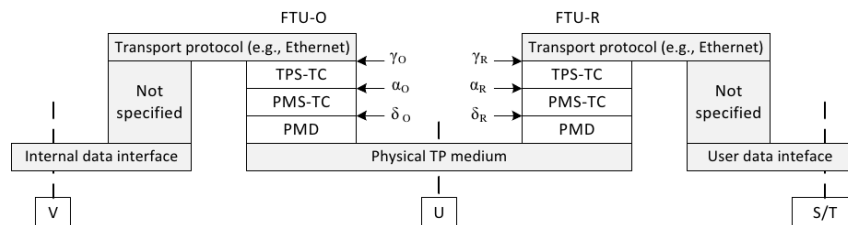


Figura 2.10: Modelo de referência do protocolo do plano de dados. Fonte: *Draft* do padrão, julho de 2013.

Similar ao modelo de referência de protocolos das gerações DSL anteriores, o modelo mostrado na Figura 2.10 indica as interfaces e suas subdivisões para comunicação entre os transceptores e com os enlaces adjacentes. Acima do meio físico, uma seção *application-invariant* – isso é, independente de e indiferente à aplicação utilizando os serviços do enlace –, é formada pelas subcamadas *physical medium dependent* (PMD) e *physical media specific part of the transmission convergence* (PMS-TC).

A subcamada PMD é responsável pela codificação, decodificação, modulação, demodulação e sincronismo dos símbolos; incluindo, possivelmente, cancelamento de eco e equalização e, sob determinação das camadas superiores, a implementação de *bit loading*,

ajuste de densidade espectral de potência (PSD), etc. Já a camada PMS-TC, logo acima, contém os serviços de *framing* e seu sincronismo, a estratégia de codificação de canal (FEC) adotada no padrão, detecção de erros, *interleaving* e *scrambling*.

A seção dependente da aplicação é formada pela subcamada *transport protocol specific transmission convergence* (TPS-TC), responsável pela configuração do tamanho do *data transfer unit* e demais parâmetros e unidades de gerenciamento como controle de fluxo e adaptação da taxa de bits. Essa subcamada é importante pois é destinada à conversão do protocolo de transporte da camada superior (por exemplo, Ethernet) ao formato unificado usado pela PMS-TC.

Por se tratar de documento do ITU-T com status de *draft*, as informações reproduzidas na Seção 2.2 podem ser mudadas ao longo do processo de padronização e, portanto, devem ser tomadas como material de referência apenas para as contribuições descritas nos capítulos desse trabalho. É importante notar também que a introdução dada aqui tem como objetivo principal a localização das funcionalidades pertinentes aos sistemas DSL, das quais o restante desse trabalho procura focar apenas a camada física.

2.2.2 A camada física do padrão G.fast

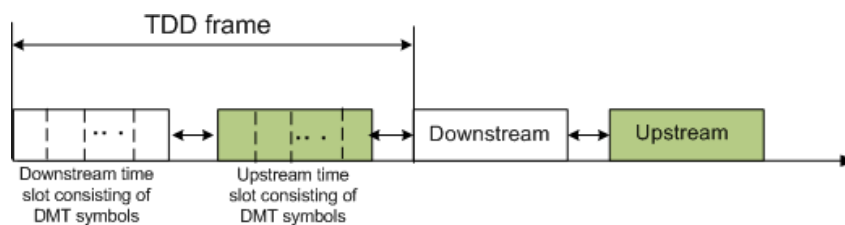
Antes de entrar em detalhes da transmissão no meio físico, é importante sumarizar as características do padrão G.fast em comparação com a geração anterior, VDSL2. Na Tabela 2.1 pode-se notar as principais diferenças entre os padrões: a largura de banda, pelo menos três vezes maior; o espaçamento entre tons, aumentado para que a complexidade da FFT se mantivesse relativamente constante; um menor número máximo de bits por tom, o que também contribui com a diminuição da complexidade e para acomodar possíveis limitações nos conversores de alta velocidade; e, também notável, a menor potência total de transmissão.

A importante mudança para S-TDD, além de facilitar a (re)configuração da taxa de assimetria, suporta também funcionamento em *low-power* ou *sleep mode*, permitindo o balanceamento entre taxa de transmissão e consumo de energia. A Figura 2.11 mostra o agrupamento dos símbolos DMT, a divisão entre *downstream* e *upstream* e a formação de um *frame*, que deve ter a duração de, no máximo, 36 símbolos DMT; resultando em latência unidirecional menor que 1 ms [27].

O modelo de referência da subcamada PMD é mostrado na Figura 2.12. No caminho de transmissão, os dados presentes na interface δ são passados a um codificador de símbolos, que contém os seguintes algoritmos:

Tabela 2.1: Comparação entre parâmetros do G.fast e VDSL2.

	G.fast (106/212 MHz)	VDSL2 (17/30 MHz)
Duplexação	S-TDD	FDD
Modulação	DMT	DMT
Espaçamento entre tons	51.75 kHz	4.3125 kHz/8.625 kHz
Número de tons	2048/4096	4096
Bits por tom	0.5 - 12 bits	0.5 - 15 bits
Codificação de canal	RS + TCM	RS + TCM
Vectoring	Linear ou não-linear	Linear
Potência de transmissão	4 dBm	14.5 dBm

Figura 2.11: *Framing* TDD adotado no G.fast.

- *Tone ordering*;
- *Trellis coding*;
- Mapeamento em constelações;
- Escalamento da constelação / ajuste de energia média;

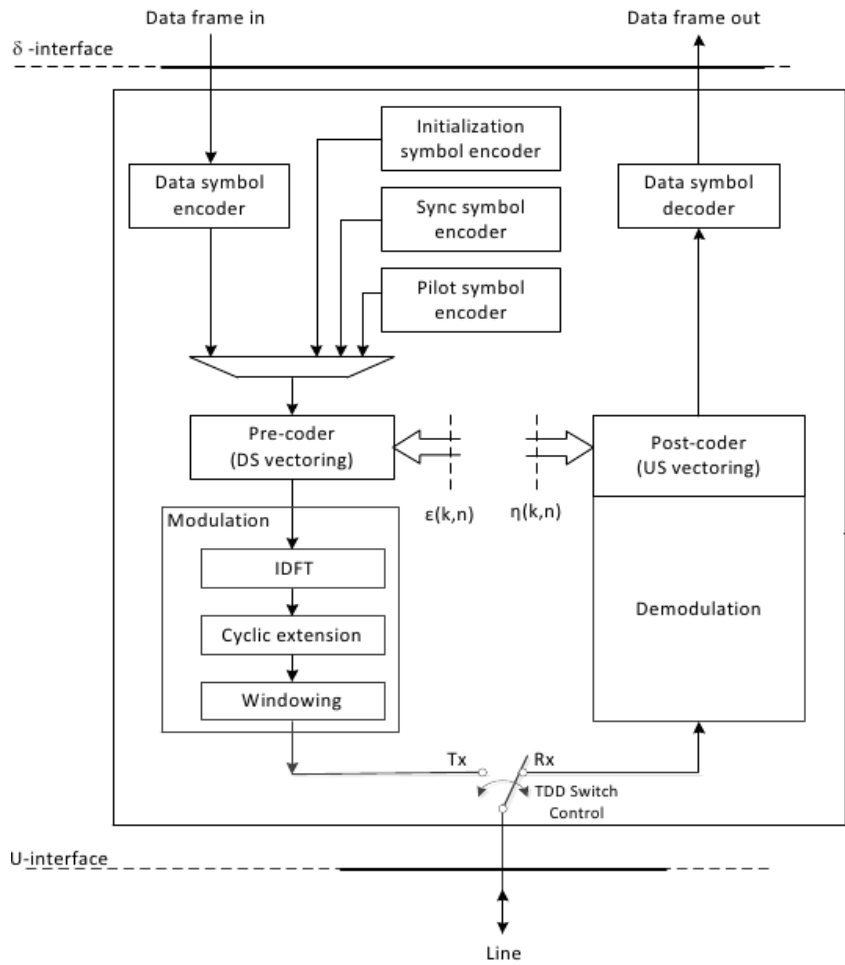


Figura 2.12: Modelo de referência da subcamada PMD. Fonte: *Draft* do padrão, julho de 2013.

Na saída dos codificadores de símbolos, uma representação do símbolo DMT no domínio da frequência está pronta. Caso o transceptor em questão seja o FTU-O, esse caminho corresponderá ao *downstream* e haverá um passo de precodificação para a implementação do *vectoring*, utilizando as interfaces ε para comunicação entre transceptores. Porque não há essa comunicação entre NTs, no transceptor FTU-R a saída dos codificadores é ligada diretamente ao modulador.

No modulador, a transformada de Fourier discreta inversa (IDFT) é aplicada para se obter uma representação do símbolo DMT no domínio do tempo e, em seguida, a extensão cíclica é aplicada, importante para a proteção contra ISI e para permitir o janelamento necessário para *spectrum shaping*.

Como é típico em recomendações xDSL, a demodulação é deixada a critério do vendedor, tipicamente executando o processo de remoção de extensão cíclica e aplicação da

transformada de Fourier discreta (DFT), seguida por equalização no domínio da frequência. Caso o caminho de recepção seja no *upstream*, haverá uma etapa de “pós-codificação” para que o *vectoring* seja implementado. Finalmente, um decodificador de símbolos transforma a representação no domínio da frequência em dados binários no formato requerido pela interface δ .

Na subcamada superior à PMD, chamada PMS-TC, é implementada a codificação de canal. Inicialmente, no caminho de transmissão, o *Scrambler* embaralha os bits de entrada da mesma forma que no padrão VDSL2. Em seguida, codificação Reed-Solomon é aplicada à entrada embaralhada, adicionando redundância que será utilizada no receptor para realizar detecção e correção de erros. Finalmente, o *interleaver* entrelaça os *bytes* das palavras-código para evitar que erros em rajada excedam a capacidade de correção do algoritmo Reed-Solomon aplicado [29]. O modelo de referência da subcamada PMS-TC é mostrado na Figura 2.13.

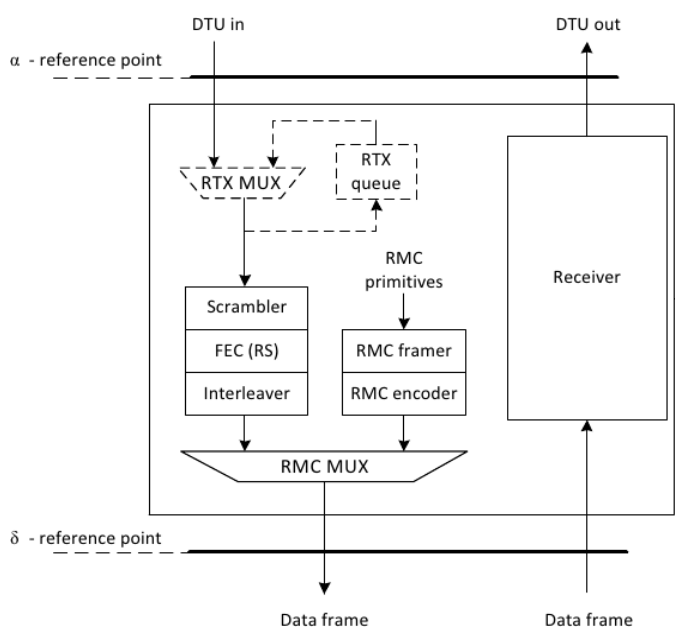


Figura 2.13: Modelo de referência da subcamada PMS-TC. Fonte: *Draft* do padrão, julho de 2013.

2.2.2.1 Procedimento de inicialização

Para que o enlace G.fast possa ser utilizado, algumas tarefas iniciais devem ser executadas pelos transceptores:

- Definição do modo de operação (taxa de assimetria, parâmetros básicos de modulação, etc.);

-
- Sincronização (alinhamento do *clock* de amostras e dos símbolos);
 - Estimação do canal e cancelamento de *crosstalk* entre as linhas ativas no grupo e as linhas entrantes;
 - Envio de parâmetros de transmissão para o FTU-R, como máscaras de PSD, bandas de radiofrequência a ser desligadas, taxas de dados em cada direção, etc.;
 - Identificação do ruído; e
 - Negociação do tamanho do *data transfer unit* (DTU), parâmetros de codificação de canal, *bit loading* e *gain scaling*.

Essas tarefas são realizadas em fases específicas da inicialização do enlace. Em primeiro lugar, ocorre um ***handshake*** padronizado pela *Recommendation ITU-T G.994.1* [30], no qual o modo de operação será negociado. Depois disso, uma etapa de ***Channel Discovery*** segue, e pela primeira vez mensagens definidas pelo padrão G.fast são trocadas, iniciando a estimação de canal e configurando os parâmetros de modulação e o uso das frequências. Finalmente, na fase de ***Channel Analysis & Exchange***, sequências predefinidas são transmitidas e, ao fim dessa fase, o *bit loading* e *gain scaling* negociados são aplicados para que se inicie o ***Showtime***, a fase em que os dados das camadas superiores são passados pelo enlace.

Capítulo 3

SIMULAÇÃO DA CAMADA FÍSICA DO SISTEMA G.FAST

Um possível problema que pode afetar a adoção de sistemas G.fast é a existência de *bridged taps* em redes residenciais. *Bridged taps* podem reduzir significativamente a taxa de transmissão por causa do aumento da atenuação e da dispersão do canal. Isso pode dificultar, por exemplo, a autoinstalação a ponto de requerer *truck rolls*, i.e. uma visita do pessoal de serviços das empresas de telecomunicação para instalar os equipamentos, modems, *splitters*, remover *bridged taps*, etc., tornando mais cara a implantação.

É, portanto, importante conhecer os impactos das *bridged taps* na capacidade do sistema. Para padronizar os experimentos e guiar o processo de especificação, o ITU-T define topologias de cabeamento como referências para implantação [31]. Essas topologias foram usadas para simular os cenários usando um simulador de transceptor DMT no domínio do tempo. Tal simulador foi desenvolvido utilizando o *software* livre GNU Radio.

GNU Radio é um *framework* flexível primariamente usado para *software-defined radio* (SDR), e fornece uma infraestrutura de processamento de sinais na forma de blocos de processamento que pode ser estendida com implementações específicas. No GNU Radio, trabalha-se com um *flow graph*, i.e. um grafo de fluxo de processamento de sinais, cujas funções de processamento são implementadas em C++, para maior desempenho, e o grafo em si é descrito em linguagem Python, para facilitar a reconfiguração [32].

3.1 SIMULADOR DE PHY G.FAST NO DOMÍNIO DO TEMPO

Para avaliar a capacidade do sistema levando em consideração a presença de *bridged taps*, um simulador de transceptor DMT chamado **gr-fast** foi desenvolvido. Em sistemas G.fast, um prefixo cíclico de curta duração deve ser usado para minimizar o *overhead*, uma vez que o período do símbolo DMT é pequeno (em torno de $20 \mu\text{s}$). Por causa disso, a taxa de transmissão pode ser limitada por ISI caso o prefixo cíclico não seja longo o suficiente para cobrir a dispersão do canal.

A ISI, que resulta de prefixo cíclico insuficiente, é simulada com precisão usando um simulador no domínio do tempo. Nesse estudo, resultados de capacidade do sistema são obtidos usando diferentes tamanhos de prefixo cíclico em diferentes topologias, e busca-se maximizar a capacidade em uma determinada topologia, observando inclusive as taxas possíveis em determinados comprimentos de cabo.

3.1.1 Etapas de simulação

Como, no padrão G.fast, o uso de *vectoring* é mandatório, as simulações descritas nesse capítulo não levam em conta o *crosstalk*, e a capacidade na linha é avaliada isoladamente. Isso diminui a quantidade de blocos necessários para se obter resultados significativos, e facilita a implementação do procedimento de inicialização.

Nas simulações adotadas, o enlace é inicializado em três etapas. Na etapa de **Deteção**, o receptor busca o preâmbulo enviado pelo transmissor no sinal recebido em meio ao ruído usando correlação cruzada em uma janela deslizante. Na etapa de **Treino-mento do equalizador**, os coeficientes do equalizador no domínio da frequência (FEQ) são atualizados, executando uma estimação do canal usando os símbolos de treinamento enviados pelo transmissor e previamente conhecidos no receptor. A inicialização termina com a etapa de **Estimação de SNR**, em que o receptor, ainda utilizando símbolos previamente conhecidos, calcula a razão sinal-ruído em cada tom para que se possa calcular a tabela de *bitloading*.

Ao fim da inicialização, a etapa de **Teste de BER** dá continuidade à simulação. Nela, o transmissor envia sequências aleatórias de bits em 1 milhão de símbolos DMT. Durante esse processo, acumula-se o número de bits decodificados erroneamente e, ao final, estima-se a taxa de erro de bits (BER) do sistema em determinado cenário.

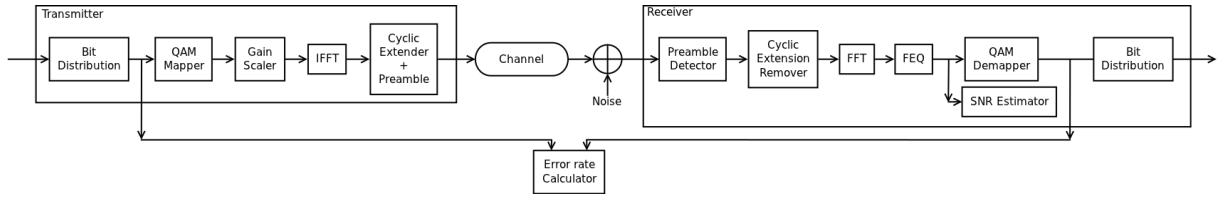


Figura 3.1: Diagrama de blocos do simulador.

3.1.2 Teoria e implementação

Na Figura 3.1, o diagrama de blocos da cadeia DMT do simulador é mostrado. Começando pelo transmissor, os bits são distribuídos a diferentes tons de acordo com uma tabela de *bitloading*. Depois, *gain scaling* é aplicado ao símbolo no domínio da frequência para controlar a PSD do sinal. Depois da IFFT, um bloco adiciona a extensão cíclica e o pré-amplificação utilizado no procedimento de inicialização.

Saindo do transmissor, o sinal é convoluído com a resposta ao impulso do canal da topologia simulada e ruído branco gaussiano é adicionado. Depois disso, no receptor, um bloco de detecção de pré-amplificação identifica o início do sinal transmitido, repassando as amostras ao removedor de extensão cíclica. Em seguida, após a FFT, o FEQ realiza a equalização cancelando a distorção de fase e amplitude do canal, e o demapeador QAM decodifica os subsímbolos em cada tom nas sequências de bits mais prováveis. Finalmente, as sequências de bits são concatenadas para se extrair os *bytes* da mensagem.

A n -ésima amostra do i -ésimo símbolo DMT após o bloco IFFT pode ser expressa como

$$x_R^{(i)}(n) = \sum_{k=0}^{2N-1} g(k) \cdot X_R^{(i)}(k) \cdot e^{j2\pi kn/2N} \quad (3.1)$$

sendo $X_R^{(i)}(k)$ o subsímbolo QAM no tom k e $g(k)$ o fator de *gain scaling*. Por causa da simetria hermitiana usada para gerar símbolos reais, $X_R^{(i)}(2N - k) = X_R^{(i)*}(k)$ e $g(2N - k) = g(k)$ para $k = 1, 2, \dots, N - 1$. Além disso, $X_R^{(i)}(N) = 0$ pois a frequência de Nyquist não é populada.

Além do prefixo cíclico, também se utiliza o sufixo cíclico no simulador. O sufixo cíclico normalmente é utilizado para sincronização dos sinais no *downstream* e *upstream* para minimizar as interferências entre suas bandas quando se usa FDD [7]. Porque o G.fast é um sistema TDD, essa interferência não é significativa. Entretanto, as topologias de referência usam modelos de canal que não são causais, o que dificulta a detecção. Para contornar esse problema, um sufixo cíclico é usado para cobrir a parte não-causal dos canais simulados.

A partir dos modelos de canal, utiliza-se a IFFT na função de transferência para gerar a resposta ao impulso $h(n)$ da linha, de modo que o sinal recebido pode ser modelado como

$$y(n) = x(n) * h(n) + w(n). \quad (3.2)$$

sendo $x(n)$ o sinal transmitido e $w(n)$ o ruído branco gaussiano, com ‘*’ denotando a operação de convolução.

3.1.2.1 Aquisição de *timing*: detecção de preâmbulo

Antes de transmitir os dados, um preâmbulo é enviado para facilitar a busca pelo início do sinal recebido. O detector de preâmbulo implementa uma detecção baseada em limiar usando uma correlação cruzada em janela deslizante com uma sequência previamente conhecida [33].

O preâmbulo é formado por dois símbolos DMT. No primeiro, os tons pares são carregados usando constelação 4-QAM com uma sequência pseudoaleatória obtida de um *linear feedback shift register* (LFSR) de 13 bits. No segundo símbolo, sequências pseudoaleatórias similares são usadas para popular todos os símbolos.

Assumindo que $r(m)$ é a amostra recebida no instante m , $p(i)$ é a i -ésima amostra do preâmbulo que tem P amostras no total, e T é o limiar de detecção; o detector calcula o produto interno $S(d)$ entre as últimas P amostras recebidas e o preâmbulo previamente conhecido,

$$S(d) = \sum_{i=0}^{P-1} r(d+i-P+1) \cdot p(i). \quad (3.3)$$

O prefixo cíclico do primeiro símbolo DMT do preâmbulo começa no instante d_0 se

$$S(d_0) \geq T \quad \text{and} \quad S(d) < T, \quad d < d_0. \quad (3.4)$$

Pode-se, então, pular as amostras do preâmbulo para iniciar a etapa seguinte de inicialização, que realiza o treinamento do equalizador.

3.1.2.2 Equalização no domínio da frequência

O FEQ é basicamente utilizado para compensar os efeitos do canal, i.e. desvio de fase e atenuação, invertendo os coeficientes do canal. Após o FEQ, o sinal $R(k)$ no tom

k pode ser escrito como

$$R(k) = F(k) (g(k)T(k)H(k) + W(k) + I(k)) \quad (3.5)$$

sendo $F(k)$ o coeficiente do FEQ no tom k , $T(k)$ o sinal transmitido e $H(k)$ a função de transferência do canal. $W(k)$ denota o componente de ruído devido à adição de ruído branco gaussiano e $I(k)$ denota o componente de ISI devido ao prefixo cíclico insuficiente [25]. Idealmente, $F(k) = 1/g(k)/H(k)$.

Para calcular $F(k)$, o algoritmo *normalized least mean squares* (NLMS) [34] foi implementado:

$$F^{(i+1)}(k) = F^{(i)}(k) + e_k^{(i)*} \cdot R^{(i)}(k) \cdot \frac{\mu}{|R^{(i)}(k)|^2} \quad (3.6)$$

sendo $e^{(i)}(k) = T^{(i)}(k) - F^{(i)*}(k) \cdot R^{(i)}(k)$ o erro no tom k entre o i -ésimo símbolo transmitido e recebido e μ o passo do LMS, com $(\cdot)^*$ denotando o complexo conjugado. Ao fim de uma sequência relativamente longa de mil símbolos de treinamento, o algoritmo NLMS converge para os coeficientes desejados.

3.1.2.3 Estimação de SNR

Depois de treinado o FEQ, inicia-se o processo de estimação de SNR. Uma sequência pseudoaleatória obtida de um LFSR com polinômio $x^{13} + x^{12} + x^{11} + x^8 + 1$ é enviada por L símbolos DMT com $g(k) = 1$. A SNR em cada tom é estimada usando *modulation error ratio* (MER) [35]:

$$\text{SNR}(k) = \frac{P(k)}{\hat{\sigma}^2(k)} \quad (3.7)$$

sendo $\hat{\sigma}^2(k) = \frac{1}{L} \sum_{i=0}^{L-1} |R^{(i)}(k) - T^{(i)}(k)|^2$ uma estimativa da potência do ruído e $P(k) = 1$ a potência dos símbolos de treinamento enviados.

3.1.2.4 Bitloading

Uma vez que a estimação de SNR tenha acabado, executa-se o algoritmo Levin-Campello de *discrete bit loading* [36] para solucionar o problema de maximização de taxa [37]: dado um limite de energia E_T , o algoritmo retorna uma N -tupla $\mathbf{b} \in \mathbb{Z}_+^N$ que otimamente distribui os bits entre N tons.

O algoritmo usa o conceito de *energia requerida* $e_k(b)$ para transmitir b bits de informação no tom k . Porque é igualmente válido usar relações de SNR onde se lê *energia*

no algoritmo, faz-se

$$e_k(b) = \frac{\varepsilon(b)}{\text{SNR}(k)} \cdot \frac{\gamma_m}{\gamma_c} \quad (3.8)$$

sendo $\varepsilon(b)$ a SNR precomputada necessária para se atingir uma taxa de erro de bit previamente definida como alvo para constelações 2^b -QAM [38] em canal AWGN. Na razão seguinte, γ_m é a margem de SNR do sistema e γ_c é o ganho de código total no sistema [1].

Para se encontrar a solução discreta ótima, as seguintes desigualdades devem ser válidas:

$$\Delta e_k(b_k) \leq \Delta e_j(b_j + \beta) \quad \forall k, j = 0, 1, 2, \dots, N - 1, \quad (3.9)$$

$$0 \leq E_T - \sum_{k=1}^N e_k(b_k) < \min_j \Delta e_j(b_j + \beta), \quad (3.10)$$

sendo β a granularidade da informação, i.e., a menor diferença (em número de bits) entre duas constelações, e

$$\Delta e_k(b) = \begin{cases} e_k(b) - e_k(b - \beta) & b \geq \beta \\ e_k(b) - e_k(0) & b < \beta \end{cases} \quad (3.11)$$

a energia incremental requerida para se transmitir b bits no tom k em comparação com a energia requerida para se transmitir $b - \beta$ bits no mesmo tom.

Aplicando-se iterativamente a Equação (3.9) a uma N -tupla, obtém-se uma distribuição de bits $b_k^{(e)}$ que é *efficient*, i.e. nenhum movimento de bits de um tom a outro reduz a energia total do símbolo. O vetor de *gain scaling* correspondente $g^{(e)}(k)$ é calculado fazendo-se

$$g^{(e)}(k) = \sqrt{e_k(b_k^{(e)})}. \quad (3.12)$$

De $b_k^{(e)}$ e $g^{(e)}(k)$, iterativamente removendo-se bits dos tons que mais consomem energia até que a Equação (3.10) seja satisfeita obtém-se uma distribuição de bits $b_k^{(t)}$ que é *E-tight*, i.e. nenhuma unidade de informação adicional pode ser enviada sem violação do limite de energia E_T estipulado. O vetor de *gain scaling* $g^{(t)}(k)$ é calculado da mesma forma que $g^{(e)}(k)$.

O par *E-tight* $b_k^{(t)}$ e $g^{(t)}(k)$ é usado como o número de bits e fator de *gain scaling* no tom k , respectivamente, quando o algoritmo de Levin-Campello converge.

3.1.3 Verificação da implementação

Para verificar a implementação dos algoritmos, executou-se teste de BER usando canais retirados de medições de cabo de comprimento 200 m, espessura 0.5 mm e banda de 200 MHz. A avaliação é feita comparando a BER alvo com a BER simulada. A Figura 3.2 mostra a resposta ao impulso do canal medido, obtida utilizando-se a transformada de Fourier discreta inversa nas medições, que por sua vez representam o ganho (complexo) do canal em cada tom analisado. Pode-se observar que as amostras antes do pico são pequenas, mas não nulas, o que corresponde à parte não-causal do canal – um artefato devido às condições imperfeitas de medição. Para ilustrar a detecção de preâmbulo, o ponto de detecção também está presente na figura, obtido superpondo-se o instante em que se ultrapassa o limiar de detecção com a resposta ao impulso do canal. Visivelmente, essa detecção não cobre todas as amostras não-causais. Entretanto, isso pode ser facilmente resolvido com algumas amostras no sufixo cíclico – nesse caso, 25 amostras são suficientes.

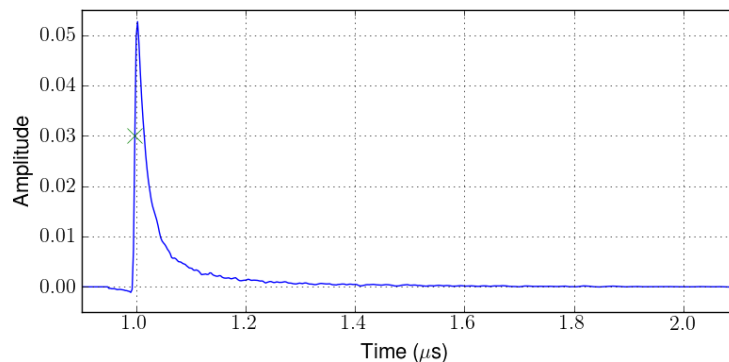


Figura 3.2: Resposta ao impulso do cabo de 200 m obtido com medições.

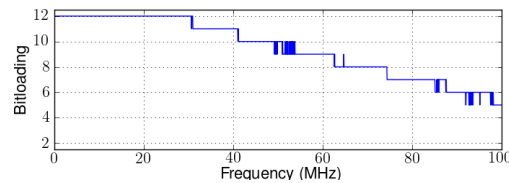
A simulação utilizou 2048 tons e um espaçamento de 48.84 kHz^1 entre eles, tendo, portanto, uma largura de banda de sistema de 100 MHz. Além disso, utilizou-se um número mínimo de 1 bit e um número máximo de 12 bits por tom; nenhum mecanismo de FEC, margem de SNR de 0 dB e uma BER alvo de 10^{-7} para se calcular a tabela de *bitloading*. Finalmente, o ruído de fundo foi configurado em -140 dBm/Hz e o sistema utilizou uma PSD de transmissão fixa em -76 dBm/Hz com limite de energia de 0 dB por tom relativo à PSD.

A Figura 3.3 compara as tabelas de *bitloading* de dois casos extremos, um quando usando uma longa extensão cíclica ($5.09 \mu\text{s}$) e outro usando uma extensão curta ($0.39 \mu\text{s}$). A extensão longa cobre mais de 99.99% da energia do canal e portanto a ISI é negligível.

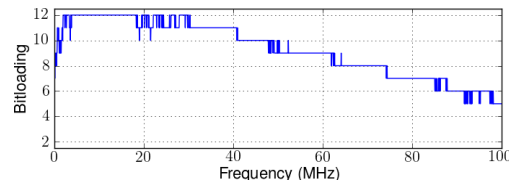
¹Esse valor difere do espaçamento apresentado na Tabela 2.1 por ter sido obtido da largura de banda do canal, 200 MHz, e não por ter sido derivado do espaçamento utilizado nas gerações anteriores.

Como esperado, devido à ISI, o *bitloading* até 4 MHz foi significativamente reduzido, mostrado na Figura 3.3b, em contraste com a situação em que se usa a extensão longa.

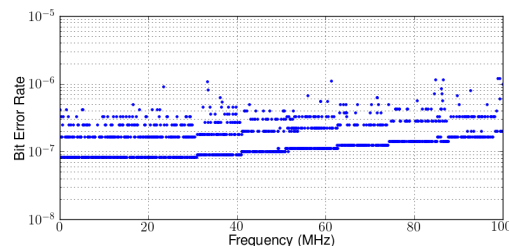
A BER atingida com extensão cíclica longa é mostrada na Figura 3.3c, alcançando uma média de $1.138 \cdot 10^{-7}$, e a mesma métrica para o caso da extensão curta é mostrada na Figura 3.3d, com uma média de $7.911 \cdot 10^{-8}$. Ambos os resultados foram muito próximos da BER alvo, mostrando que o simulador corretamente atinge o desempenho desejado tanto em situações de ISI quanto em situações sem interferência. Deve-se notar que a BER em alguns tons é zero e portanto não visível no gráfico, e o pequeno aumento da BER nas altas frequências pode ser causado pelo treinamento do FEQ prejudicado, uma vez que nesses tons a SNR é mais baixa. No geral, isso pode ser ignorado para o que se propõe esse estudo.



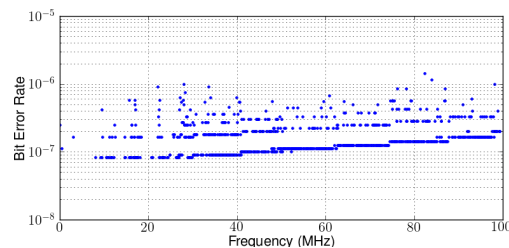
(a) Tabela de *bitloading* com extensão cíclica longa ($5.09 \mu\text{s}$).



(b) Tabela de *bitloading* com extensão cíclica curta ($0.39 \mu\text{s}$).



(c) BER com extensão cíclica longa ($5.09 \mu\text{s}$).



(d) BER com extensão cíclica curta ($0.39 \mu\text{s}$).

Figura 3.3: Resultados para *bitloading* e taxa de erro de bits por tom.

3.2 SIMULAÇÃO DE CAPACIDADE EM CENÁRIOS DE REFERÊNCIA

Com foco em avaliar o impacto de diferentes extensões cíclicas na capacidade do sistema G.fast em cenários de referência do ITU-T, apresenta-se nessa seção a abordagem, configurações e resultados obtidos nas simulações.

3.2.1 Abordagem

A abordagem básica é usar o simulador para calcular a tabela de *bitloading* assumindo parâmetros de sistema como margem de SNR, tamanho da extensão cíclica, ganho de código, bits de redundância, etc., sem de fato transmitir dados do usuário. Tendo verificado o simulador na Seção 3.1.3, não é necessário executar o teste de BER para avaliar a capacidade. Isso economiza um tempo considerável de simulação e portanto acelera o estudo da capacidade do sistema.

Após obter a tabela de *bitloading*, a capacidade R do sistema pode ser calculada como

$$R = \frac{R_c}{T_{\text{sym}}} \sum_{k=0}^{N-1} b_k \quad (3.13)$$

sendo $T_{\text{sym}} = T_s(L_{\text{CE}} + 2N)$ e T_s os períodos do símbolo DMT e de amostragem, respectivamente; R_c o ganho total de código e L_{CE} a duração da extensão cíclica, em amostras. Se Reed-Solomon e *trellis coding* são usados, como em VDSL2, o ganho total de código pode ser calculado como $R_c = R_{\text{RS}} \cdot R_{\text{TC}}$.

As taxas de código para Reed-Solomon (R_{RS}) e *trellis coding* (R_{TC}) são dadas por

$$\begin{aligned} R_{\text{RS}} &= K_{\text{RS}}/N_{\text{RS}} \\ R_{\text{TC}} &= 1 - \frac{1}{N_{\text{L}}} \sum_{k=0}^{N-1} t_{\text{ovh}}(k) \\ t_{\text{ovh}}(k) &= \begin{cases} 0.5/b_k & b_k > 0 \\ 0 & \text{otherwise} \end{cases} \end{aligned} \quad (3.14)$$

sendo N_{RS} e K_{RS} o tamanho da palavra código e do bloco de dados Reed-Solomon, respectivamente, N_{L} o número de tons *carregados*, i.e. com *bitloading* não-nulo; e $t_{\text{ovh}}(k)$ leva em conta o *overhead* imposto pelo *Wei's 4-dimensional 16-state trellis code*.

3.2.2 Topologias de referência do ITU-T simuladas

Na Figura 3.4, as topologias de referência implementadas são mostradas, com comprimentos de cabo atingindo até 250 metros. As topologias 1 e 2 são baseadas no modelo de cabo CAD55 para um *drop cable* típico no Reino Unido [39]. A topologia 2 tem uma pequena *bridged tap* no cabeamento residencial, enquanto a topologia 1 é uma linha reta.

As topologias 3 and 4 são baseadas no modelo de cabo ANSI TP1 para cabos de cobre 26 AWG normalmente usados em redes dos EUA como *drop cables* [40]. A topologia 4 representa o pior canal, com 5 *bridged taps* conectadas ao mesmo ponto.

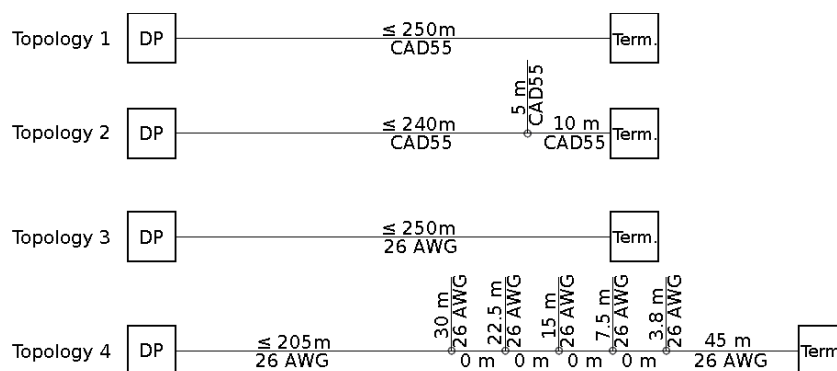


Figura 3.4: Topologias do ITU-T simuladas.

Deve ser notado que, porque ambos os modelos CAD55 e ANSI TP1 são baseados no modelo BT0, que é não-causal, a detecção de preâmbulo teve que cobrir os precursores não-causais, o que aumentou a extensão cíclica requerida. Isso significa que, em cabos reais, a extensão utilizada pode ser ainda menor já que não há essa não-causalidade.

3.2.3 Configuração da simulação

Os parâmetros de sistema usados na simulação estão listados na Tabela 3.1. Ao gerar a resposta ao impulso de canais, é importante reduzir a oscilação temporal por causa da janela retangular implícita no domínio da frequência, especialmente para cabos curtos. Para mitigar isso, o sistema é sobreamostrado duas vezes, gerando uma curva de 200 MHz, e a resposta ao impulso obtida é então filtrada usando um filtro Kaiser de 53 coeficientes e banda de transição entre 100 e 120 MHz, o que efetivamente emula também os efeitos dos filtros de transmissão e recepção. A Figura 3.5 mostra como os efeitos oscilatórios são mitigados com essa filtragem.

Tabela 3.1: Parâmetros de simulação.

Largura de banda do canal e do símbolo	200 MHz e 100 MHz
Espaçamento em frequência	48.84 kHz
Número de tons carregados e tamanho da FFT	2048 e 8192
PSD de transmissão e do ruído de fundo	-76 e -140 dBm/Hz
Limite de energia	0 dB por tom relativo à PSD
Codificação de canal	Reed-Solomon + <i>Trellis coding</i>
Reed-Solomon	N/R = 255/16
<i>Trellis coding</i>	<i>overhead</i> de 0.5 bit por tom
Ganho de código total	5 dB
Margem de SNR	6 dB
<i>Gap</i> de SNR	9.75 dB
<i>Bitloading</i>	1-12 bits
Duração da extensão cíclica	0.4, 0.8, 1.2, 1.6, 2.0, 2.4 μ s
Tamanho do cabo (do DP ao NT)	50, 100, 150, 200, 250 m

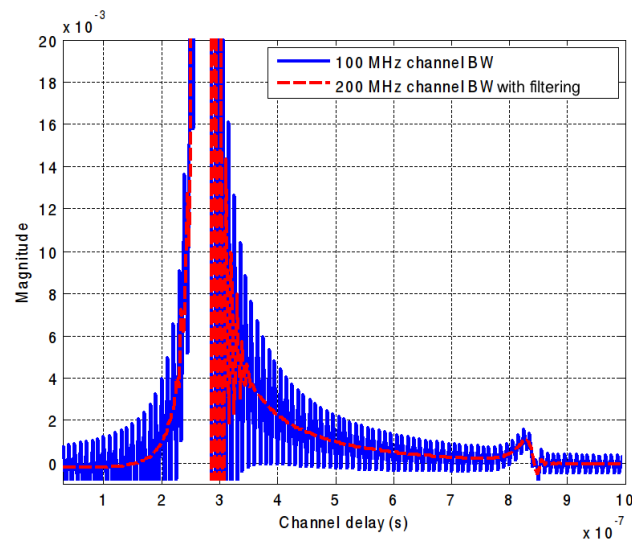


Figura 3.5: Resposta ao impulso de cabo CAD55 com comprimento de 50 metros gerado com largura de banda de 100 MHz sem filtragem adicional versus canal filtrado usando largura de banda de 200 MHz. Nota: os coeficientes foram normalizados e deslocados para comparação visual.

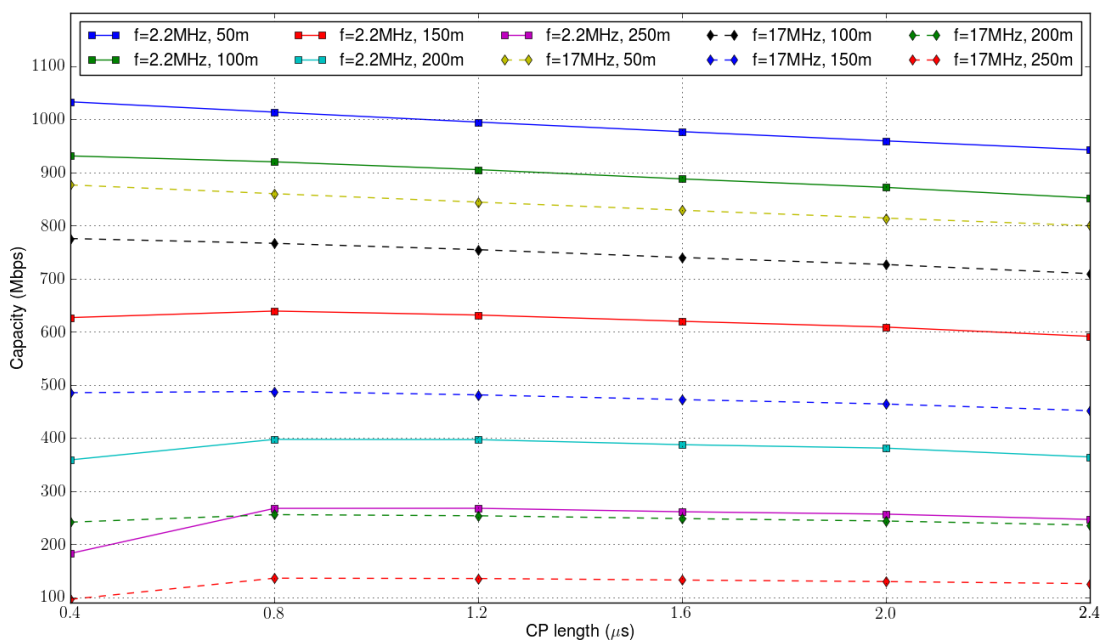
3.2.4 Tamanho ótimo da extensão cíclica

Os resultados de capacidade nas topologias 1 e 4 são mostrados na Figura 3.6, sendo f a frequência inicial do sistema G.fast, implementando compatibilidade espectral com sistemas ADSL2/2+ ($f=2.2$ MHz) e VDSL2 ($f=17$ MHz, $f=30$ MHz). A Figura 3.6a mostra que a capacidade diminui com o aumento da extensão cíclica, tendência mostrada também pelas topologias 2 e 3, ocultadas por brevidade. Isso indica que a dispersão do canal é pequena para linhas diretas (topologias 1 e 3) e linhas com uma *bridged tap* curta (topologia 2) e, portanto, poucas amostras na extensão cíclica são suficientes (i.e. $0.8 \mu\text{s}$). Contudo, como mostrado na Figura 3.6b, a topologia 4 requer extensões muito mais longas (por exemplo, $2.0 \mu\text{s}$) em virtude da maior dispersão do canal causada pela condição extrema de 5 *bridged taps*.

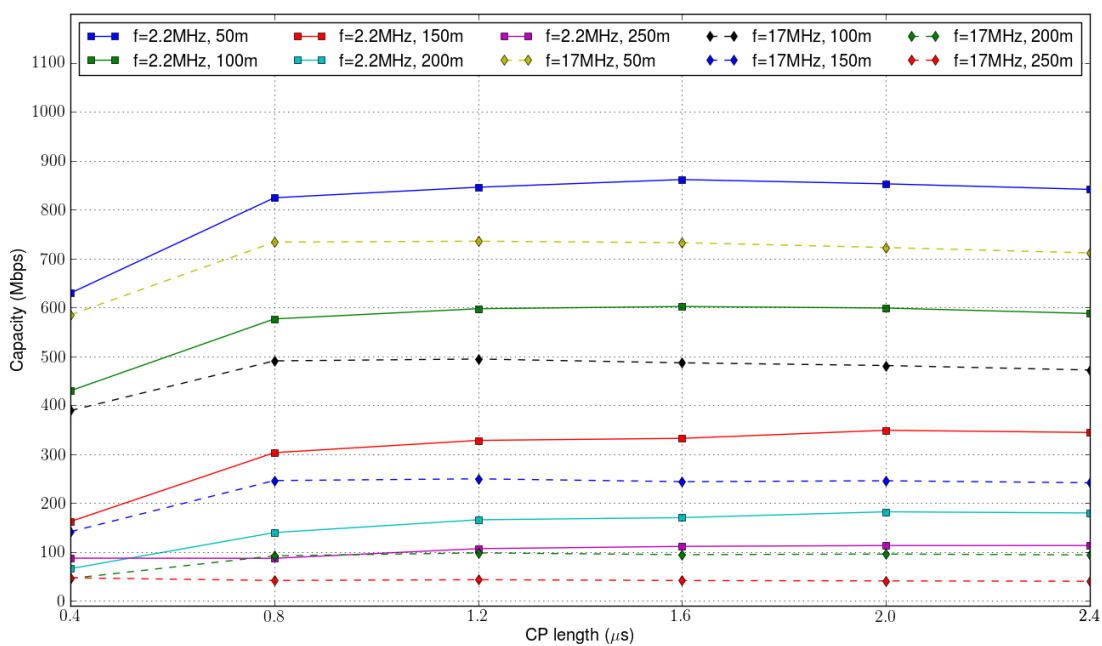
Tabela 3.2: Tamanhos ótimos para extensão cíclica (em microssegundos).

Frequência inicial	2.2 MHz	17 MHz	30 MHz
Topologia 1	0.8	0.8	0.8
Topologia 2	0.8	0.8	0.8
Topologia 3	0.8	0.8	0.8
Topologia 4	2.0	1.6	1.2

A Tabela 3.2 sumariza o tamanho ótimo da extensão cíclica para diferentes frequências iniciais de operação e diferentes topologias. Ela mostra que essa duração está no intervalo de 0.8 a $2.0 \mu\text{s}$, correspondendo a um *overhead* de 3.76% a 8.9% , respectivamente. Para a topologia 4, a extensão cíclica ótima diminui com o aumento da frequência inicial de operação. Isso acontece porque a ISI se concentra nas frequências mais baixas. Portanto, frequências maiores são muito menos afetadas, como mostrado também nos resultados de *bitloading* na Seção 3.1.3.



(a) Capacidade na topologia 1.



(b) Capacidade na topologia 4.

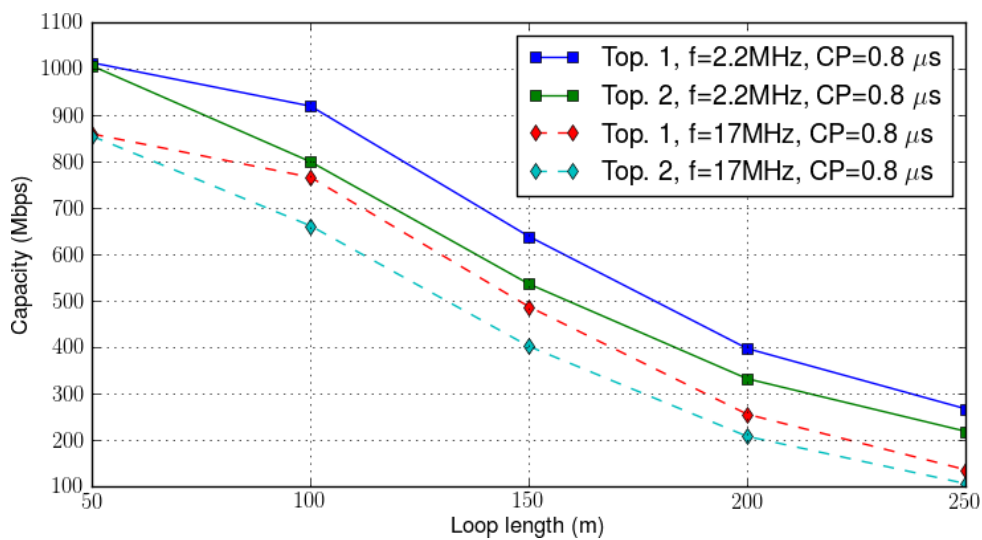
Figura 3.6: Impacto da duração da extensão cíclica na capacidade nas topologias 1 e 4.

3.2.5 Estudo *rate-reach*

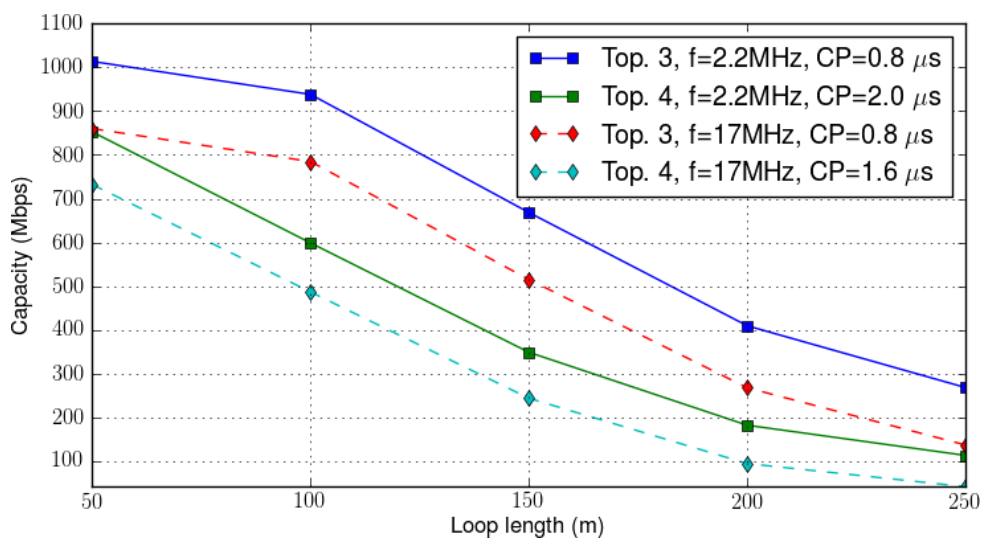
A Figura 3.7 mostra os resultados *rate-reach* (taxa de transmissão de bits versus comprimento do cabo) com as extensões cíclicas ótimas para cada topologia simulada e com frequências iniciais de operação de 2.2 e 17 MHz. A habilidade de atingir altas taxas de transmissão do sistema G.fast, mesmo com cenários prejudicados por *bridged taps*, é visível. Comparadas com as topologias 1 e 3, as *bridged taps* nas topologias 2 e 4 reduzem a taxa de transmissão significativamente. Por exemplo, a 100 m com frequência inicial 2.2 MHz, uma redução de mais de 300 Mbps acontece na topologia 4, e redução de mais de 100 Mbps na acontece topologia 2. Entretanto, o sistema ainda é capaz de atingir em torno de 350 Mbps a 150 m na topologia 4 (que é a pior delas). Para topologias de linha direta (1 e 3), o sistema pode até atingir 400 Mbps a 200 m e em torno de 900 Mbps a 100 m.

Ainda na Figura 3.7, iniciando a operação em 17 MHz para coexistir com sistemas VDSL2 no mesmo *bundle*, há alguma redução na taxa de transmissão. Ainda assim, o sistema é capaz de manter mais de 400 Mbps a 150 m nas topologias 1 a 3. Para a topologia 4, mais de 250 Mbps pode ser atingido a essa distância.

A partir desses resultados, pode-se concluir que uma extensão cíclica ótima encontra-se entre 0.8 e 2.0 μ s para diferentes topologias. Sistemas G.fast são robustos a *bridged taps* ponto de prover mais de 500 Mbps a 150 m e, mesmo no pior caso, mais de 350 Mbps pode ser atingido nessa distância. Isso indica que, evitando os caros *truck rolls*, o custo de implantação de G.fast é consideravelmente menor que outras abordagens para banda larga.



(a) Diminuição da capacidade nas topologias 1 e 2.



(b) Diminuição da capacidade nas topologias 3 e 4.

Figura 3.7: Impacto do aumento do comprimento do cabo na capacidade.

Capítulo 4

PROTÓTIPO DE TRANSCCEPTOR BASEADO EM G.FAST

Continuando com a pesquisa em sistemas G.fast, apresenta-se nesse capítulo a segunda parte da plataforma DSL em desenvolvimento. Além do *software* de simulação, usado para desenvolver e testar algoritmos DMT, uma plataforma de prototipagem baseada em DSP foi desenvolvida para se implementar os algoritmos em um ambiente mais realístico para um modem G.fast e, portanto, utilizar os cabos de cobre disponíveis nos laboratórios.

A plataforma de prototipagem foi inicialmente dividida em duas partes para isolar as implementações, como mostra a Figura 4.1. O Módulo Digital é responsável pelo processamento de sinais da cadeia DMT, enquanto um *analog front-end* (AFE) contém a circuitaria necessária para filtragem e acoplamento com a linha telefônica e conversão das amostras.

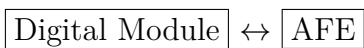


Figura 4.1: Visão geral da plataforma de prototipagem.

O AFE, que não faz parte desse trabalho, tem o diagrama de blocos mostrado na Figura 4.2. A cadeia de recepção é formada por um amplificador de baixo ruído (LNA), um filtro analógico anti-*aliasing*, um amplificador de ganho programável (PGA) e um conversor analógico-digital (ADC). Na cadeia de transmissão, um conversor digital-analógico (DAC) gera a forma de onda, que é filtrada por um filtro analógico anti-imagem e amplificada pelo *line driver*. Finalmente, fazendo a conexão com o canal, um circuito de híbrida acopla o AFE à linha telefônica permitindo a comunicação bidirecional.

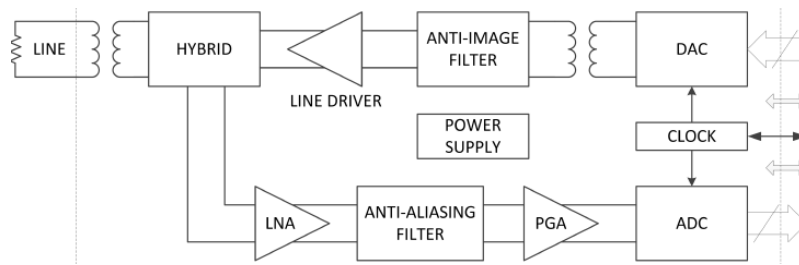


Figura 4.2: Diagrama de blocos do AFE.

Um importante requerimento da plataforma de prototipagem é executar os processos de modulação e demodulação respeitando os limites de tempo impostos pelos conversores no AFE. Esses requerimentos dependem da largura de banda do sistema e ditam vários parâmetros da plataforma, como latência do barramento [41], taxa de sinalização, capacidade computacional, etc. Nesse trabalho, optou-se por uma arquitetura híbrida [42] para a plataforma de prototipagem.

Para implementar o protótipo, o módulo DSP escolhido foi um System-on-Chip (SoC) Texas Instruments *multicore*, TMS320C6670, que trabalha tanto com ponto fixo quanto com ponto flutuante. No SoC, cujo diagrama de blocos é mostrado na Figura 4.4, vários subsistemas estão disponíveis para algoritmos comumente utilizados em implementações de camada física de sistemas de comunicação, dos quais o Coprocessador FFT (FFTC), o periférico Serial RapidIO [43] (SRIO) e o *Network Coprocessor* (NETCP) são os mais importantes no momento.

Além do componente DSP da plataforma, o componente FPGA Xilinx Virtex-6 é usado para interfaceamento com o AFE e para implementação de algoritmos mais apropriados para seu paradigma de funcionamento. Por causa da opção por trabalhar com kits de avaliação para acelerar o desenvolvimento, uma placa intermediária chamada “Breadboard” foi confeccionada no laboratório para conectá-los, como mostra o diagrama do protótipo na Figura 4.3.

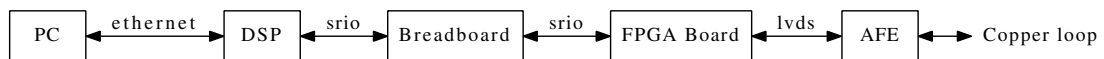


Figura 4.3: Diagrama de blocos do protótipo.

Em uma visão geral, o protótipo consiste de um modem DMT baseado em DSP que troca dados de usuário vindos de um enlace Ethernet com os conversores no AFE usando uma interface LVDS convertida pelo FPGA para o *interconnect* RapidIO. A primeira versão do modem tem como objetivo implementar uma cadeia DMT completa sem codificação de canal para a banda do sistema G.fast.

4.1 CARACTERÍSTICAS DO DSP

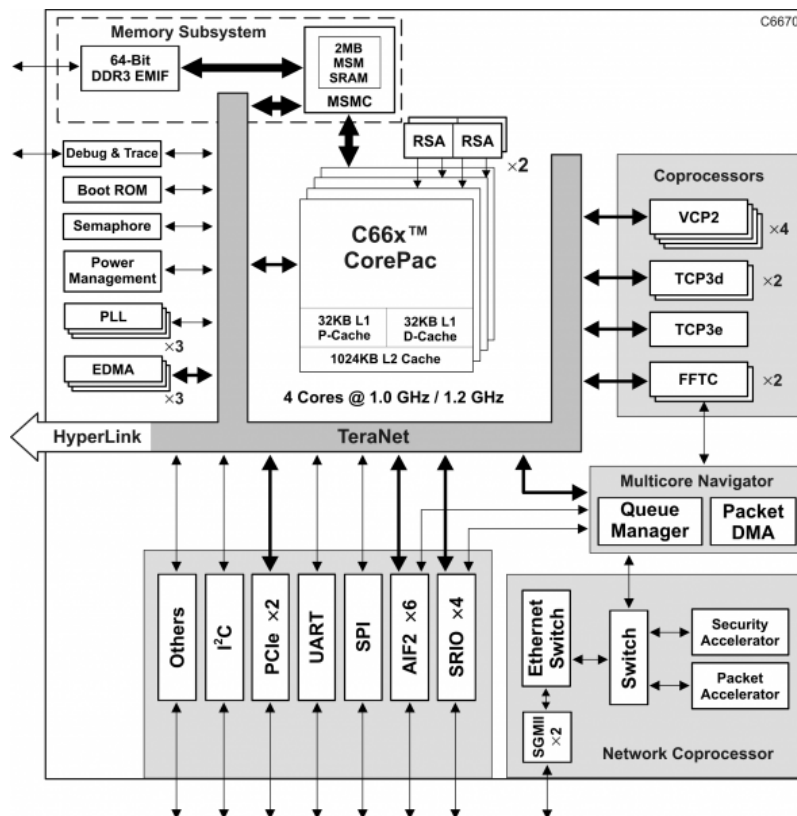


Figura 4.4: Diagrama de blocos do TMS320C6670. Fonte: *Texas Instruments TMS320C6670 Data Manual*.

O kit de avaliação do TMS320C6670 escolhido para desenvolvimento é a placa TMDXEVM6670L (EVM). As características mais importantes são as interfaces de alta capacidade disponibilizadas no conector AMC, à direita da Figura 4.5 – em que está presente, por exemplo o *interconnect* RapidIO para comunicação com o FPGA. Também de sumária importância são o *chip* de camada física ethernet, ENET PHY, que provê uma interface gigabit; e o chip USB-JTAG para programação e depuração.

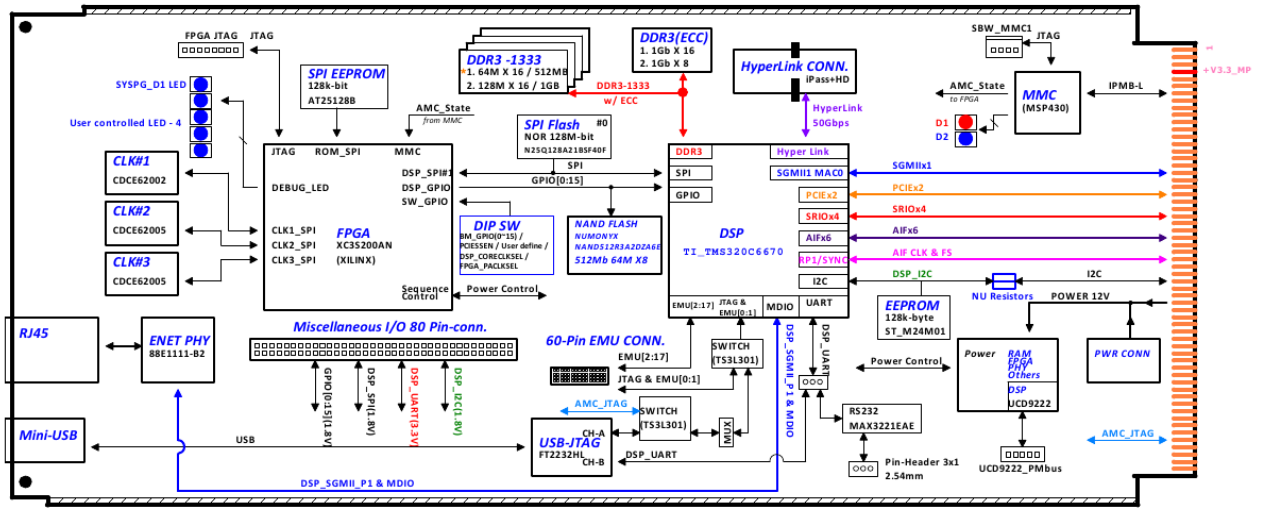


Figura 4.5: Diagrama de blocos da placa TMDXEVM6670L. Fonte: *Texas Instruments TMDXEVM6670L Technical Reference Manual*.

O C6670 pertence a uma família de dispositivos chamada *Keystone Architecture*. O principal mecanismo de transporte nessa família é o *Multicore Navigator*, um gerenciador de filas de pacotes com um despachador implementado em *hardware* que direciona as tarefas de cada periférico envolvido na operação. O componente principal do *Multicore Navigator* é o *Packet DMA* (PKTDMA), um mecanismo de *direct memory access* (DMA) especializado para o uso desses pacotes e presente em praticamente todos os coprocessadores do SoC. O diagrama de blocos do *Multicore Navigator* é mostrado na Figura 4.6.

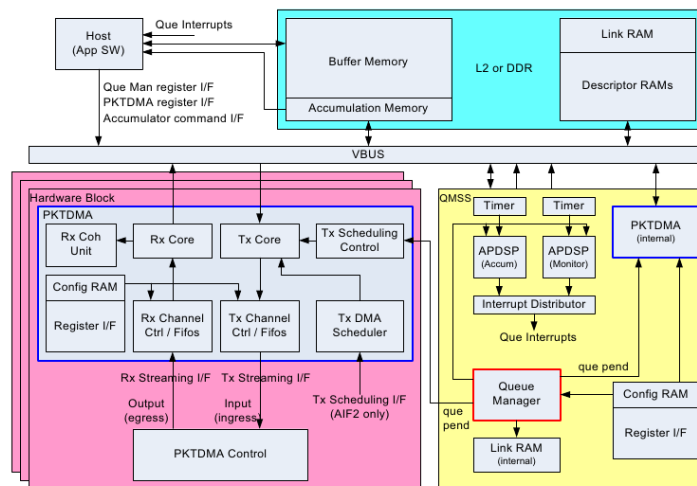


Figura 4.6: Diagrama de blocos do *Multicore Navigator*. Fonte: *Texas Instruments Multicore Navigator User Guide*.

O *Multicore Navigator* e suas interfaces PKTDMA são uma peça central na implementação do protótipo desse trabalho, uma vez que possibilitam comunicação *thread-safe* e sem *overhead* entre os núcleos de processamento, coprocessadores e periféricos. O mecanismo de funcionamento é baseado em *descritores*, que são estruturas contendo ponteiros para os dados que se deseja enviar e informações de operação, sendo enfileirados sob gerenciamento do *Queue Manager Subsystem* (QMSS) como mostra a Figura 4.7. Nota-se que há diferentes tipos de descritores, dependendo da aplicação: pode-se desejar uma alocação de memória independente (*host descriptors*) ou utilizar a própria estrutura de dados para armazenar as informações (*monolithic descriptors*).

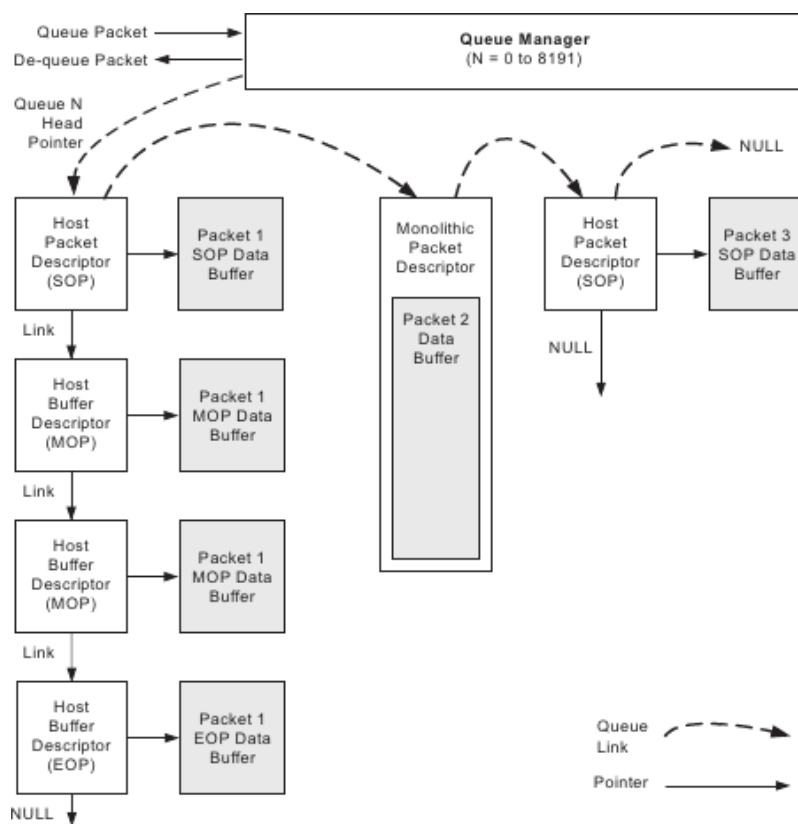
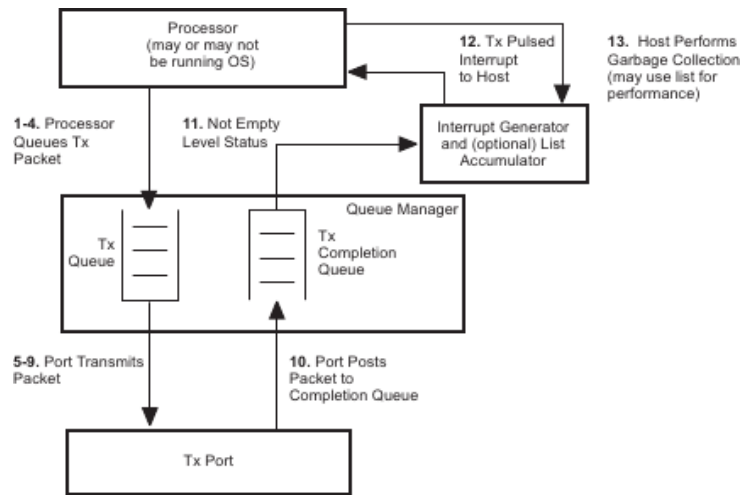


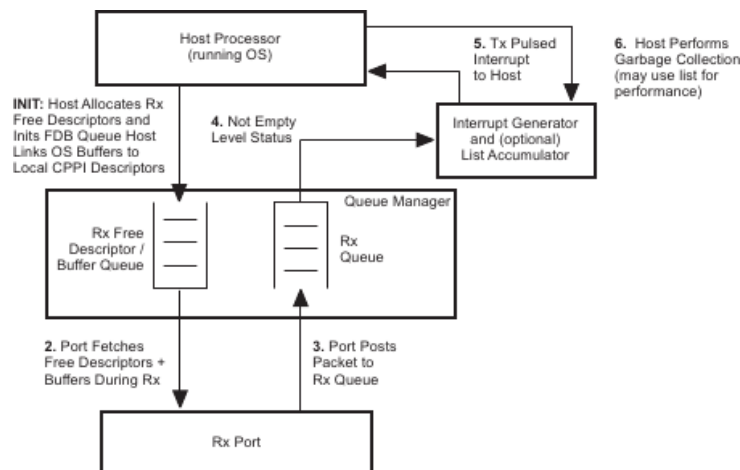
Figura 4.7: Enfileiramento de descritores no *Multicore Navigator*. Fonte: *Texas Instruments Multicore Navigator User Guide*.

O gerenciador de filas, QMSS, utiliza as informações auxiliares nos descritores para encaminhar as tarefas para os elementos correspondentes. Mais ainda, o QMSS emprega o conceito de *filas de operação* e *filas livres*. Como mostra a Figura 4.8, os mecanismos de transmissão e recepção podem ser configurados para acontecer automaticamente. No caso da transmissão, inicia-se o processo com a aplicação rodando no DSP adicionando um descritor a uma determinada fila de operação. Isso, por sua vez, faz com que o *hardware* (*Tx Port*) inicie a cópia da informação armazenada, a fim de transmiti-la para

o seu destino. Ao fim da cópia, o descritor é redirecionado para uma fila *livre*, podendo ser novamente reaproveitado pela aplicação.



(a) Passos para transmissão.



(b) Passos para recepção.

Figura 4.8: Transmissão e recepção de pacotes no QMSS.

O mecanismo de recepção funciona de forma similar, diferenciando-se apenas pela entidade que inicia a operação. Nesse caso, o *hardware* (*Rx Port*), recebendo uma requisição vinda de algum *Tx Port*, utiliza um dos descritores disponíveis na fila livre, popula a área de dados com a informação recém-obtida, e enfileira o descritor na fila de recepção. Isso torna essa informação disponível para uso da aplicação. Em ambos os casos, pode-se configurar a geração de interrupções para que a aplicação seja notificada dos eventos.

Esse mecanismo é central na implementação do protótipo. Em virtude da presença universal do *Multicore Navigator* no SoC, escolheu-se a operação por filas por permitir a tradução direta da cadeia de processamento de sinais que geralmente se utiliza ao im-

plementar o *line coding* DMT. Mais ainda, também fornecido pela Texas Instruments é o sistema operacional de tempo real (RTOS) SYS/BIOS, um *software* livre disponível sob licença BSD. O SYS/BIOS contém bibliotecas diversas para configuração e utilização dos periféricos e coprocessadores, bem como primitivas de sincronização (semáforos, eventos, portas, etc.), gerenciamento de memória (*heaps*, alocação de memória, proteção de memória, etc.), gerenciamento de *threads* (interrupções de *hardware* e de *software*, *Tasks*, etc.), entre outros, todas de interesse para a implementação do protótipo. A arquitetura da aplicação será descrita com mais detalhes nas seções seguintes.

4.2 ARQUITETURA E DESENVOLVIMENTO DA APLICAÇÃO

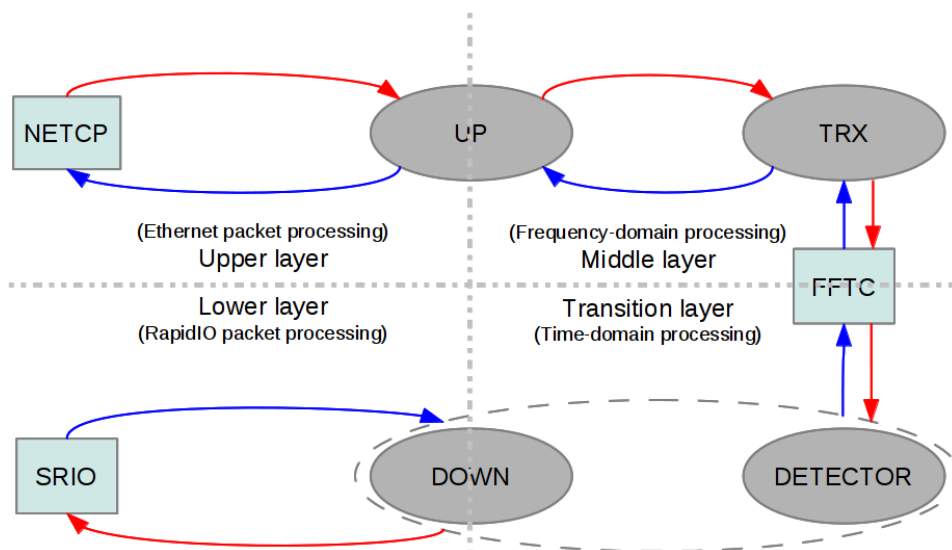


Figura 4.9: Arquitetura da aplicação para o protótipo.

A atual versão da arquitetura da aplicação é mostrada na Figura 4.9. Nesse diagrama, as elipses representam núcleos de processamento no SoC e retângulos representam coprocessadores ou periféricos; arestas vermelhas representam a cadeia de transmissão e arestas azuis representam a cadeia de recepção. A elipse tracejada englobando os núcleos na *Transition layer* significa que os papéis desses núcleos pode mudar, dependendo do comportamento das *threads* e do RTOS ao se fazer o interfaceamento com o periférico SRIO.

Upper layer

O núcleo UP faz o interfaceamento com o NETCP e é o primeiro componente na tradução de quadros Ethernet em quadros G.fast, na cadeia de transmissão; e na remontagem dos últimos nos primeiros na cadeia de recepção. Também nesse núcleo roda o algoritmo de *bit slicing* e segmentação de quadros, responsável por quebrar os *bytes* de entrada em sequências de bits possíveis de se enviar nas constelações QAM adotadas. *Forward error correction* e, mais tarde, adaptação de taxa são os próximos candidatos para implementação nesse núcleo.

Middle layer

O núcleo TRX contém a maioria do *line coding* DMT no domínio da frequência, a saber, mapeamento QAM, *gain scaling* e a montagem de pacotes a ser transformados pelo FFTC, na cadeia de transmissão; e multiplicação do FEQ e demapeamento QAM na cadeia de recepção. A maioria das computações serão realizadas por bibliotecas externas como a DSPLIB, também disponibilizada pela Texas Instruments sob licença BSD.

A adição de prefixo cíclico pode ser feita pelo próprio FFTC “em *batch*” (i.e. vários símbolos DMT por requisição) caso sua duração seja um múltiplo de 4, o que geralmente é suportado. Isso diminui o *overhead* da aplicação, uma vez que é realizado diretamente no *hardware* do SoC. Na cadeia de recepção, porém, a remoção de prefixo cíclico é limitada a um bloco por requisição e, portanto, é feita na aplicação.

Transition layer e Lower layer

Os núcleos DOWN e DETECTOR são destinados a processamento no domínio do tempo. Independente da divisão de tarefas entre esses núcleos, na cadeia de recepção os pacotes SRIO devem ser quebrados e dois possíveis caminhos de execução existem: (1) os símbolos DMT no domínio do tempo são passados ao FFTC, ou (2) o código de detecção do sinal está rodando, sem qualquer interação com as camadas superiores. Essa divisão é interessante para se explorar mecanismos de *idle mode*, já que durante a detecção ou em períodos em que não é necessário o processamento no domínio da frequência, as camadas superiores podem ser desligadas. Na cadeia de transmissão, o símbolo no domínio do tempo é passado ao periférico SRIO em pacotes de 2048 amostras após qualquer processamento necessário, por exemplo *windowing* e *overlapping*.

4.3 TESTES DE UTILIZAÇÃO DO *INTERCONNECT* SRIO

Em virtude da escolha do *interconnect* SRIO como a interface de alta velocidade para as amostras no domínio do tempo, é importante assegurar o uso correto do enlace no protótipo. Assim, testes extensivos foram conduzidos para garantir que a taxa de transmissão e requerimentos de *timing* sejam atingidos pelos elementos processadores. Em uma primeira etapa, chamada *Connectivity Tests*, a funcionalidade de transporte de dados foi implementada e resultados preliminares foram obtidos para o SRIO e, separadamente, para o enlace Ethernet.

Depois dos *Connectivity Tests*, a nova versão do protocolo de comunicação entre DSP e FPGA foi implementada, diminuindo o *overhead* de sinalização, o que permitiu que requerimentos de *timing* menos estritos fossem possíveis no FPGA. Os resultados do *benchmark* usando a taxa de dados mais alta do periférico, i.e. 4x 5.0 Gbaud, são mostrados na Figura 4.10, em que é possível notar que o uso atual do enlace excede os requerimentos de banda para operação a 400 MSPS de 16 bits cada, totalizando 6.4 Gbps.

Esses resultados foram obtidos contando-se o número de ciclos necessários para se executar uma *bulk transaction*, i.e. uma transmissão de 4096 *bytes*, e a cada iteração no gráfico calculou-se a média do número de ciclos necessários para se completar 100000 transações. Uma média global, próxima de 7.9 Gbps, mostra que o requerimento é atingido com uma margem em torno de 1.5 Gbps.

Na figura, também é possível notar um *outlier* em destaque, que possivelmente resultou de uma operação de *link recovery*. Isso, combinado com os relatórios de “*error encountered*” emitidos pelo periférico durante a negociação da taxa, sugere que o enlace pode, às vezes, se tornar instável a ponto de ser necessária uma operação de *recovery*, que é mais severa que uma retransmissão. Entretanto, os resultados apontam que essa instabilidade temporária não prejudica severamente o mecanismo de transporte.

Para explorar ainda mais a capacidade do enlace, pode-se diminuir a sinalização pela metade, e fazendo-se isso foi possível atingir uma taxa de 10.9 Gbps usando transações NWRITE, que não requerem respostas de *acknowledgement*. Essa taxa está próxima ao *benchmark* publicado pela Texas Instruments, reportando 13.08 Gbps com a mesma transação. Suspeita-se, então, que o fator limitante nesse caso seja apenas a menor otimização do código. De fato, o protocolo atual requer respostas de *acknowledgement* para identificar problemas na comunicação entre DSP e FPGA, mas os resultados servem para indicar que a utilização completa do enlace depende apenas de investimento na otimização do código atual, de modo que a aplicação DMT pode tomar como garantia

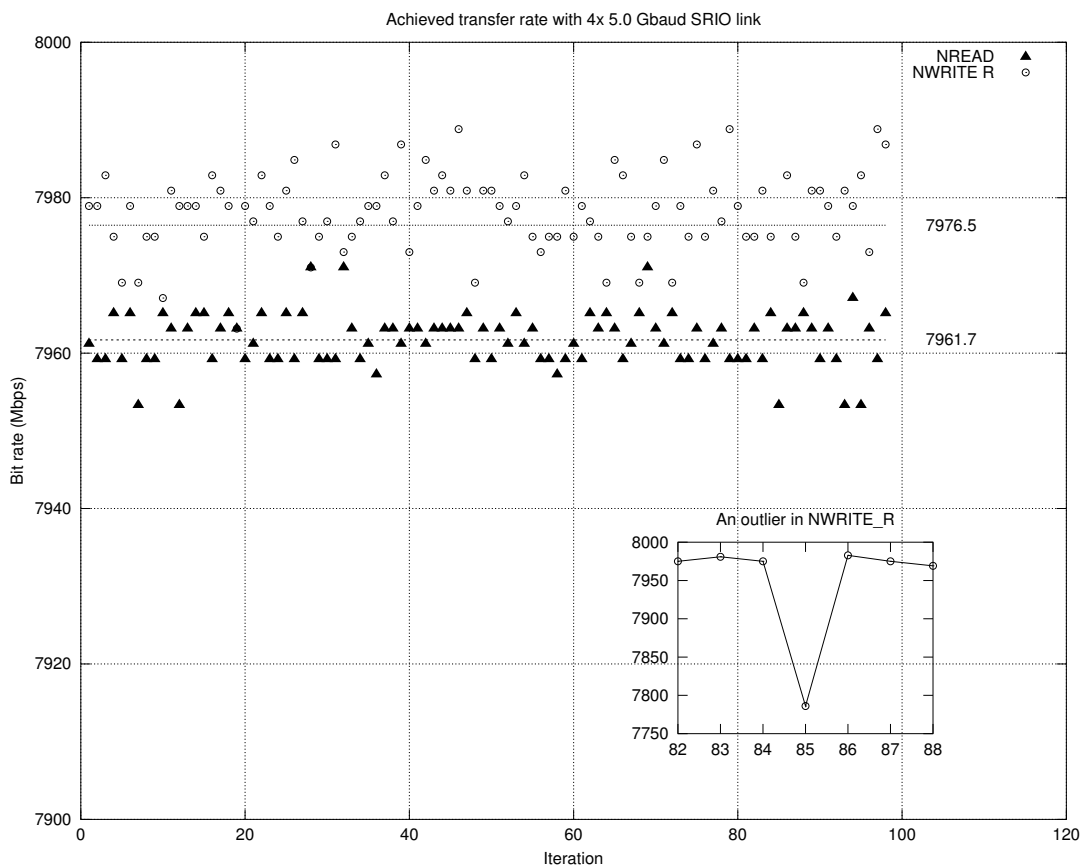


Figura 4.10: Resultados de *benchmark* para transações NREAD e NWRITE_R.

a entrega das amostras aos conversores no AFE.

Capítulo 5

CONSIDERAÇÕES FINAIS

Nesta dissertação, as etapas de padronização, simulação e implementação de um sistema de comunicação de banda larga foram apresentadas. A banda larga de quarta geração, atualmente em processo de especificação pelo ITU-T na forma do padrão G.fast, promete aumentar as taxas de transmissão do serviço residencial, bem como explorar a já existente malha de cobre em aplicações como *backhaul* de redes sem fio. No Capítulo 1, indicadores sociais e um histórico das gerações de sistemas DSL mostraram a evolução da tecnologia ao longo dos anos.

O conceito e características desse novo sistema foram apresentadas no Capítulo 2. Seguindo as gerações de DSL anteriores, o novo padrão utiliza as mesmas técnicas de codificação de canal, cancelamento de *crosstalk* por *vectoring* e proteção contra ruído impulsivo. Diferente dos predecessores, o G.fast utiliza duplexação S-TDD e largura de banda na casa das centenas de MHz, aproveitando a tendência de encurtamento dos cabos de cobre para atingir uma maior taxa de transmissão.

No Capítulo 3, uma primeira avaliação desse sistema foi feita na forma de simulações no domínio do tempo para avaliação de capacidade. Levando em conta as ramificações tipicamente existentes nas topologias próximas à área residencial, chamadas *bridged taps*, uma campanha de simulação foi definida para confirmar a robustez do sistema em cenários de referência definidos pelo ITU-T. Após o desenvolvimento de um simulador adequado à tarefa, concluiu-se o sucesso da implantação a altas taxas residenciais assumindo-se o cancelamento total do *crosstalk* usando a coordenação entre os transceptores. Essa contribuição foi publicada:

I. Almeida, A. Klautau, and C. Lu, “Capacity Analysis of G.fast Systems Via Time-domain Simulations,” in *Communications (ICC), 2013 IEEE International Conference on*, 2013.

A segunda contribuição desse trabalho, mostrada no Capítulo 4, está relacionada à implementação dos algoritmos necessários ao sistema em questão em um ambiente mais realístico. Para isso, uma arquitetura híbrida de DSP e FPGA foi utilizada para se criar um modem DMT baseado no padrão, cujo projeto segue em desenvolvimento.

Tanto o simulador quanto o protótipo fornecem uma plataforma de transmissão digital de sinais multiportadora que pode ser estendida para várias aplicações e avaliações de desempenho. Como trabalhos futuros, propõe-se (1) a implementação de diferentes algoritmos de *forward error correction*, como LDPC e Reed-Solomon; (2) algoritmos de adaptação de taxa para reconfiguração dinâmica da taxa de transmissão; (3) precodificação não-linear e mecanismos de *power loading* para menor consumo de energia pelos transceptores; (4) algoritmos de cancelamento de eco e de NEXT; (5) transmissão não-contínua e algoritmos de *sleep mode*, também visando a diminuição do consumo de energia, entre outros. Todos possíveis em ambos os ambientes uma vez que a implementação básica esteja terminada.

No Brasil, a distribuição desigual da cobertura de banda larga entre os estados deve ser discutida. A exclusão digital é atualmente um tema de debates entre organizações multilaterais, governos e empresas, e acredita-se que o desenvolvimento de ferramentas nacionais para tecnologias de ponta como mostrado nesse trabalho possam sanar alguns dos problemas que prejudicam a implantação de redes de comunicação de qualidade em lugares fora do eixo econômico nacional, por exemplo, o próprio Pará, que goza de pouco mais de 10% de seus habitantes com acesso a banda larga residencial. Esse trabalho se propõe, então, a prover à academia e à sociedade ferramentas para pesquisa e desenvolvimento de tecnologias que possam melhorar a qualidade de vida.

Referências Bibliográficas

- [1] T. Starr, J. M. Cioffi, and P. J. Silverman, *Understanding Digital Subscriber Line Technology*, Prentice-Hall, 1999.
- [2] “Plan Ceibal,” <http://www.ceibal.org.uy/>, Último acesso em agosto de 2013.
- [3] “Fundação para o Desenvolvimento da Educação,” <http://www.fde.sp.gov.br/>, Último acesso em agosto de 2013.
- [4] Raul Katz, “The impact of broadband on the economy: Research to date and policy issues,” Tech. Rep., International Telecommunication Union - Telecommunication Development Sector, 2012.
- [5] Broadband Commission for Digital Development, “Broadband: A platform for progress,” Tech. Rep., International Telecommunication Union, 2011.
- [6] Broadband Commission for Digital Development, “The state of broadband 2012: Achieving digital inclusion for all,” Tech. Rep., International Telecommunication Union, 2012.
- [7] Philip Golden, Herve Dedieu, and Krista Jacobsen, *Fundamentals of DSL Technology*, Auerbach Publications, Taylor & Francis Group, 2006.
- [8] Broadband Forum, “MR-257: An Overview of G.993.5 Vectoring,” Tech. Rep., Broadband Forum, 2012.
- [9] Telebrasil, “O Desempenho do Setor de Telecomunicações no Brasil: Séries Temporais 1T13,” Tech. Rep., Telebrasil, 2013.
- [10] ITU-T, “Asymmetric Digital Subscriber Line (ADSL) transceivers,” June 1999.
- [11] ITU-T, “Asymmetric Digital Subscriber Line transceivers 2 (ADSL2),” .
- [12] ITU-T, “Asymmetric Digital Subscriber Line (ADSL) transceivers - Extended bandwidth ADSL2 (ADSL2+),” .

-
- [13] M. Monteiro, “Otimização de sistemas DSL utilizando técnicas de gerenciamento de espectro,” M.S. thesis, Universidade Federal do Pará, 2010.
- [14] ITU-T, “Very high speed digital subscriber line transceivers,” June 2004.
- [15] ITU-T, “Very high speed digital subscriber line transceivers 2 (VDSL2),” February 2006.
- [16] J. Maes, M. Guenach, K. Hooghe, and M. Timmers, “Pushing the limits of copper: Paving the road to FTTH,” in *Communications, IEEE International Conference on*, june 2012, pp. 3149–3153.
- [17] P. Ödling, T. Magesacher, S. Höst, P. O. Börjesson, M. Berg, and E. Areizaga, “The fourth generation broadband concept,” *Communications Magazine, IEEE*, vol. 47, no. 1, pp. 62–69, january 2009.
- [18] “ITU Readies New 1Gbit/s Broadband Standard,” http://www.itu.int/net/pressoffice/press_releases/2013/30.aspx, Último acesso em agosto de 2013.
- [19] I. Almeida, A. Klautau, and C. Lu, “Capacity Analysis of G.fast Systems Via Time-domain Simulations,” in *Communications (ICC), 2013 IEEE International Conference on*, 2013.
- [20] Broadband Forum, “MR-185: Next Generation Broadband Access White Paper,” Tech. Rep., Broadband Forum, 2009.
- [21] J.W. Cook, R.H. Kirkby, M.G. Booth, K.T. Foster, D.E.A. Clarke, and G. Young, “The noise and crosstalk environment for ADSL and VDSL systems,” *IEEE Commu. Mag.*, vol. 37, no. 5, pp. 73–78, May 1999.
- [22] Broadband Forum, “MR-180: Achieving quality IPTV over DSL,” Tech. Rep., Broadband Forum, 2012.
- [23] G. Ginis and J.M. Cioffi, “Vectored transmission for digital subscriber line systems,” *IEEE Commu. Mag.*, vol. 20, no. 5, pp. 1085–1104, june 2002.
- [24] ITU-T Rec. G.993.5-2010, “Self-FEXT cancellation (vectoring) for use with VDSL2 transceivers,” 2010.
- [25] W. Henkel, G. Taubock, P. Odling, P.O. Borjesson, and N. Petersson, “The cyclic prefix of OFDM/DMT - an analysis,” in *Broadband Communications, International Zurich Seminar on*, 2002, pp. 22–1–22–3.

-
- [26] Les Brown, “G.fast for FTTdp,” September 2012.
- [27] Associate Rapporteur for G.fast, “Updated draft text for G.fast - version 6.0,” ITU-T temporary document 2013-07-Q4-R20, July 2013.
- [28] PLOAM Standard G.997.1, “International telecommunications union,” 2002.
- [29] Steven A. Tretter, *Communication System Design Using DSP Algorithms*, Springer, 2008.
- [30] ITU-T, “Handshake procedures for digital subscriber line transceivers,” June 2012.
- [31] Editor for G.fast, “G.fast: Wiring topologies and reference loops,” Tech. Rep., ITU-T Q4/15 Contribution 11GS3-100, Sep 2011.
- [32] “GNU Radio,” <http://www.gnuradio.org>, Último acesso em agosto de 2013.
- [33] T.M. Schmidl and D.C. Cox, “Robust frequency and timing synchronization for OFDM,” *Communications, IEEE Transactions on*, vol. 45, no. 12, pp. 1613–1621, dec 1997.
- [34] Simon Haykin, *Adaptive Filter Theory*, Prentice Hall, 3rd edition, 1996.
- [35] Yin Wang, Zhaowu Chen, and Ke Gong, “MER performance analysis of M-QAM-OFDM with Wiener phase noise,” in *ICMMT’07*, april 2007, pp. 1–4.
- [36] Jorge Campello, “Practical Bit Loading for DMT,” *IEEE ICC’99 Proceedings*, pp. 801–805, 1999.
- [37] T. Starr, M. Sorbara, J. M. Cioffi, and P. J. Silverman, *DSL Advances*, Prentice-Hall, 2003.
- [38] ITU-T Rec. G.9960-2010, “Unified high-speed wire-line based home networking transceivers - system architecture and physical layer specification,” 2010.
- [39] BT plc, “G.fast: Cable models,” Tech. Rep., ITU-T Q4/15 Contribution 11RV-026, Nov 2011.
- [40] ANSI Standard T1.417-2003, “Spectrum management for loop transmission systems,” 2003.
- [41] Abhijit Athavale and Carl Christensen, *High-Speed Serial I/O Made Simple*, Xilinx, 2005.

- [42] Volodymyr S. Podosinov, “A Hybrid DSP and FPGA System for Software Defined Radio Applications,” M.S. thesis, Virginia Polytechnic Institute and State University, 2011.
- [43] Motorola Semiconductor Product Sector, “RapidIO: An Embedded System Component Network Architecture,” Tech. Rep., Motorola Semiconductor, 2000.