Projeto de Protocolos Utilizando HW/SW Codesign Baseado na Otimização de Desempenho por Algoritmos Genéticos

Marcio N. Miranda[1] **, Ricardo N.B. Lima**[2] **,**
Aloysio C.P. Pedroza[2] **, Antônio C. Mesquita Filho**[2]

[1]Instituto de Informática – Universidade Federal de Goiás
Bloco IMF I - sala 227 Campus II - Samambaia 74001-970 Goiânia, GO

[2]Universidade Federal do Rio de Janeiro
Centro de Tecnologia - bloco H - sala 321
Caixa Postal 68504 - Ilha do Fundão 21945-970 Rio de Janeiro, RJ

`miranda@inf.ufg.br,belem,aloysio@gta.ufrj.br,mesquita@coe.ufrj.br`

Resumo. *Este trabalho apresenta uma metodologia de HW/SW Codesign aplicada à síntese de protocolos baseada em técnicas de análise de desempenho e em algoritmos genéticos. A metodologia é aplicada a um protocolo de controle de handoff para redes ATM sem fio (WATM). Nesse processo deve-se escolher as tarefas a serem implementadas em hardware e as tarefas a serem implementadas em software, ou seja, a seleção da melhor partição HW/SW. A metodologia utiliza de forma agregada uma ferramenta de análise de desempenho, ferramentas de síntese de HW e um algoritmo genético para otimizar uma função objetivo. O objetivo da metodologia é auxiliar os projetistas de protocolos a selecionar uma partição HW/SW que forneça o melhor compromisso entre o custo e o desempenho.*

Abstract. *A design-to-hardware implementation methodology applied to the synthesis of communication protocols based on performance analysis techniques and genetic algorithms is presented. The process requires a choice between the tasks to be implemented in hardware and those to be implemented in software, i.e., the selection of the best HW/SW partition. This methodology uses a genetic algorithm to optimize an error function defined by the required performance and the implementation cost of the protocol. The protocol performance is analyzed by a modelling tool. Hardware implementations using are presented and compared to software implementations.*

1. Introdução

As redes de computadores de alta velocidade exigem protocolos de comunicação específicos. O uso da otimização em *software* (SW) na elaboração de protocolos para as atuais redes nem sempre permitem operações a altas taxas de velocidade. Um fator importante para o bom desempenho dos protocolos de alta velocidade é a utilização integrada de *hardware*(HW) e *software* (SW) durante sua implementação. Algumas operações do sistema, normalmente as menos críticas em termos de velocidade, podem ser implementadas em SW, enquanto que aquelas que necessitam de uma maior velocidade devem ser implementadas em HW. Devido à crescente demanda por protocolos com altas vazões, as soluções que utilizam HW são cada vez mais investigadas. Num roteador, por exemplo, o encaminhamento de pacotes totalmente implementado em SCs (*Standard Cells*) pode fornecer uma vazão até dez vezes maior do que a implementação inteiramente baseada em microprocessadores de propósito geral [1].

Se no desenvolvimento do projeto as partes de HW e SW são integradas numa única especificação e desenvolvidas conjuntamente, o processo é conhecido como *HW/SW codesign* [2]. O princípio do *codesign* é a realização de um projeto cooperativo baseado em ambientes específicos, HW e SW, onde pode-se verificar e simular todo o sistema em cada etapa do projeto. Esse processo permite a detecção de erros nas etapas iniciais do projeto, reduzindo o seu tempo e o seu custo.

Apesar dos tempos de execução em um HW específico serem, normalmente, muito menores que os tempos de execução em uma implementação em SW, o custo da primeira solução é maior. Portanto, deve-se estabelecer um compromisso entre o desempenho desejado e o custo de implementação para se decidir que partes deverão ser implementadas em HW e que partes deverão ser implementadas em SW. Essa etapa é denominada de **processo de partição HW/SW** e sua decisão tem impacto direto no desempenho do sistema de comunicação do qual o protocolo faz parte.

O projetista ainda dispõe de poucos recursos para auxiliá-lo no refinamento da especificação, no desenvolvimento do projeto e no processo de partição HW/SW de um dado protocolo. Diversos grupos de pesquisa vêm desenvolvendo ambientes de projeto baseados na metodologia de *codesign*. Dentre eles, pode-se citar: Ptolemy [3], LYCOS [4] e PISH [5]. Esses ambientes de *codesign* diferem na linguagem de especificação, no método de particionamento, na arquitetura alvo e nos métodos de validação utilizados.

O objetivo e principal contribuição deste trabalho é apresentar uma nova metodologia que auxilie o processo de partição HW/SW de um protocolo de comunicação, estabelecendo critérios objetivos para que a mesma seja realizada sem necessidade do projetista utilizar técnicas manuais. Essa proposta utiliza de forma conjunta uma ferramenta de análise de desempenho, ferramentas de síntese de HW em SCs (*standard cells*) e PLD (*Programmable Logic Devices*) e um algoritmo genético.

A metodologia de projeto desenvolvida neste trabalho é uma evolução de técnicas de HW/SW Codesign anteriormente propostas e é aplicada especificamente ao projeto de protocolos de comunicação. A busca por essa evolução é motivada pela inexistência de técnicas que auxiliam objetivamente o processo de partição HW/SW, limitando a exploração do espaço de soluções. Nesse caso, é útil fornecer ao projetista subsídios que o auxiliem na escolha da melhor partição HW/SW. Essa metodologia foi desenvolvida em Miranda [6] e neste trabalho é aplicada um protocolo de controle de *handoff* para uma rede ATM sem fio.

Este trabalho está estruturado da seguinte forma: a seção 2 apresenta uma descrição suscinta da metodologia de projeto, a seção 3 apresenta os resultados da aplicação da metodologia ao protocolo mencionado acima e a seção 4 apresenta os comentários finais. Em Miranda [6] pode-se analisar resultados obtidos para outros tipos de protocolos.

2. Metodologia de Projeto

Na metodologia desenvolvida neste trabalho, os protocolos são especificados, inicialmente, por um modelo de máquina de estados, onde cada transição de estado representa uma operação ou um conjunto de operações a serem avaliadas. Esta especificação serve como base para a construção do modelo da ferramenta de análise de desempenho Tangram-II [7] e para a descrição na linguagem VHDL [8] de maneira que a síntese dos circuitos de HW possa ser realizada.

A ferramenta Tangram-II auxilia o projetista no cálculo de parâmetros de desempenho tais como retardo, vazão, probabilidade de perda de mensagens, entre outros. Um algoritmo genético é utilizado para otimizar uma função objetivo desses parâmetros. As ferramentas Synopsys [9] e Altera [10] fornecem informações de custo do HW a partir dos circuitos integrados sintetizados.

Inicialmente é especificado o desempenho desejado para o protocolo, em termos da vazão, banda passante, retardo ou qualquer outra medida de interesse que possa ser obtida a partir da ferramenta de análise de desempenho aqui utilizada. O protocolo é especificado com base nas redes de Petri estocásticas [11] para que possa ser realizada uma verificação e validação da especificação através da ferramenta ARP (analisador de redes de Petri), que verifica se a rede não possui *deadlocks* e/ou *livelocks*, além de fornecer outras características importantes, como, por exemplo, se a rede é *viva* e *limitada* [12]. Uma vez verificada a especificação, é construído o modelo apropriado à análise de desempenho do protocolo, desenvolvido no ambiente e na sintaxe da ferramenta de análise de desempenho Tangram-II. O modelo desenvolvido é paramétrico, sendo os valores dos parâmetros fornecidos por um método de otimização baseado em algoritmos genéticos (AGs) [13]. As medidas de desempenho desejadas são utilizadas na composição da função objetivo que será otimizada. Isso permite encontrar as taxas de transição da rede de Petri que melhor aproximam o desempenho e custo especificados. Uma vez determinadas as taxas de transição da rede de Petri, são estabelecidos critérios objetivos para determinar a partição HW/SW.

Inicialmente, o protocolo é descrito por um modelo de máquina de estados, onde cada transição de estado representa uma operação ou um conjunto de operações a serem avaliadas. A partir desta especificação, o fluxo de projeto segue dois caminhos paralelos. No primeiro, a especificação é detalhada e o modelo é construído utilizando-se a ferramenta Tangram-II. Os parâmetros do protocolo que atendem aos requisitos de desempenho especificados são determinados pelo Tangram-II e pelo algoritmo genético. No segundo caminho são obtidas as medidas de atraso e custo através da ferramenta Synopsys, que fornece como saída um circuito lógico a partir do qual pode-se extrair as medidas de área e atraso.

2.1. Otimização do Desempenho

A partir da especificação do protocolo em máquina de estados é realizada a sua modelagem na ferramenta Tangram-II, através de um modelo com parâmetros, que são as taxas dos *eventos*. O problema inicial consiste em determinar as taxas de transição de estado que fornecem os requisitos de desempenho especificados. A determinação dessas taxas fornecem subsídios para o estabelecimento de um critério de partição.

Os valores numéricos necessários para resolver o modelo do Tangram-II são fornecidos por um método de otimização baseado em AGs que, inicialmente, gera conjuntos de valores para os parâmetros de forma aleatória. Após a substituição dos parâmetros por esses valores numéricos, o Tangram-II resolve a cadeia de Markov associada ao modelo. Estas probabilidades são utilizadas para estimar o desempenho do protocolo através do valor de uma função objetivo que é avaliada para cada conjunto de parâmetros gerado pelo AG, ou seja, para cada *indivíduo* gerado pelo AG. O valor da função objetivo fornece a *aptidão* de um determinado indivíduo para resolver o problema.

Nessa etapa inicia-se o processo de otimização, onde uma nova *população* é gerada para ser avaliada pelo Tangram-II, até que uma solução seja encontrada. As taxas encontradas como solução são utilizadas para obter uma partição HW/SW que é avaliada pelas medidas de área e atraso fornecidas pelo Synopsys. Caso a área e o atraso da partição atendam à especificação, a solução foi encontrada, caso contrário uma nova população é gerada.

2.2. Implementação em HW

A especificação do protocolo em máquina de estados serve como referência para a entrada das descrições comportamentais em VHDL para o Synopsys e o ALTERA. Cada transição de estado, contendo uma ou mais operações do protocolo, é associada a uma *entidade* em VHDL. Cada *entidade* é sintetizada, de modo a se obter as medidas de atraso e área referentes às transições de estado quando implementadas em HW.

A ferramenta Synopsys sintetiza um circuito lógico a partir da descrição comportamental de cada transição em VHDL. Esta implementação é realizada utilizando-se uma biblioteca de células padrão (*standard cells*) numa dada tecnologia. As etapas de alocação e roteamento de células e otimização são automáticas e se tornam “transparentes” para o projetista.

PLDs são circuitos integrados digitais que podem ser programados a fim de implementar funções complexas, isto é, PLDs podem implementar qualquer expressão booleana definida através de programação. A ferramenta ALTERA implementa as funções lógicas de cada transição a partir de suas descrições em VHDL.

Ao final do processo de síntese, obtém-se uma tabela contendo todas as medidas de atraso e custo de área para cada uma das transições do protocolo. Estas medidas serão utilizadas para avaliar se a partição escolhida atende à especificação de custo do HW fornecida pelo projetista e se os respectivos atrasos atendem às taxas calculadas pelo AG. Se a avaliação for positiva, o processo se encerra, caso contrário, uma nova otimização é realizada.

2.3. A Função Objetivo

De uma forma geral, a *função objetivo* ou função de erro do problema de partição HW/SW pode ser dada pela diferença entre os valores das medidas de desempenho desejadas, especificadas pelo projetista, e os valores das medidas de desempenho obtidas para um dado conjunto de parâmetros. Como os parâmetros especificados normalmente “competem entre si”, é útil atribuir-se pesos a cada um dos parâmetros que compõem a função objetivo para equalizar as possíveis diferenças de sensibilidade entre eles [14]. Essa função fornece uma medida da qualidade de uma determinada solução e através de sua avaliação pode-se selecionar aquela que satisfaz melhor os requisitos de desempenho. A forma geral da função objetivo é a seguinte:

$$\varepsilon(\lambda) = \sum_{i=1}^{n} w_i * |f_{d_i}(\lambda) - f_{o_i}(\lambda)| \quad \mathbb{R}^n \rightarrow \mathbb{R}^1 \tag{1}$$

onde:

$\lambda = (\lambda_1, ..., \lambda_n)$ - é o vetor de parâmetros da otimização;

f_{d_i} - é a i-ésima medida de desempenho desejada;

f_{o_i} - é a i-ésima medida de desempenho obtida;

w_i - são os fatores de ponderação ou normalização;

n - número de parâmetros de desempenho a serem otimizados;

A função objetivo utilizada neste trabalho é definida a partir de medidas de desempenho do protocolo, tais como vazão, retardo, probabilidade de perda, entre outras, obtidas a partir do Tangram-II. As medidas de desempenho utilizadas dependem dos requisitos de qualidade de serviço desejados. O problema de otimização consiste em determinar λ^* tal que:

$$\varepsilon(\lambda^*) = \min_{\lambda} \varepsilon(\lambda) \tag{2}$$

2.4. O Critério de Partição

Com o objetivo de auxiliar o projetista no processo de partição é fundamental estabelecer um critério que relacione as taxas calculadas pelo AG com uma determinada partição HW/SW. Quanto maior a taxa de um determinado evento maior o número de vezes, por segundo, que um determinado conjunto de operações do protocolo é executado. Deve-se considerar, a partir dos resultados obtidos resolvendo-se a cadeia de Markov associada, as probabilidades em estado estacionário de cada estado do sistema. Ou seja, mesmo que um evento tenha uma alta taxa, a probabilidade do sistema estar num estado em que o mesmo está habilitado pode ser muito baixa.

A partir das considerações realizadas acima, foi estabelecido como critério de partição HW/SW dos eventos o produto $\lambda_i * \pi$, onde:

λ_i - taxa do evento i determinada pelo AG.

π - soma das probabilidades em estado estacionário dos estados nos quais o evento i está habilitado.

O critério de partição apresentado é usado como uma estimativa inicial da partição HW/SW e fornece subsídios ao projetista para uma escolha objetiva da melhor partição. Essa partição é avaliada através das medidas de área e atraso pela ferramenta Synopsys.

2.5. Algoritmo Genético Implementado

O algoritmo utilizado neste trabalho é *híbrido*, pois mistura AGs com métodos de otimização tradicionais, que utilizam derivadas na determinação das faixas iniciais.

O algoritmo desenvolvido é um AG simples, onde o cromossomo foi codificado como um vetor de variáveis reais que representam os parâmetros (taxas dos eventos) a serem determinados. A dimensão do vetor depende do número de taxas que se deseja determinar, conforme o modelo construído para o protocolo em questão.

A população inicial é obtida por uma função que gera números reais aleatoriamente, utilizando um banco de sementes e o relógio do computador. Os valores numéricos correspondentes a cada indivíduo são transferidos pelo AG ao modelo de parâmetros do Tangram-II. Desta forma é possível executar um dos métodos numéricos de solução da cadeia de Markov existentes no Tangram-II através de uma rotina específica, que é chamada pelo AG, e determinar para cada indivíduo da população as probabilidades em estado estacionário da cadeia de Markov, obtendo-se as respectivas medidas de interesse para cada um destes indivíduos. A aptidão bruta é determinada através do valor da função objetivo, dado pela soma de erros entre os valores das medidas obtidas e os valores desejados, conforme a equação 1. Ao final do processo de otimização, o melhor indivíduo é usado para definir a partição HW/SW, a qual é avaliada pela ferramenta de síntese de HW.

3. Resultados

A metodologia apresentada foi aplicada a um protocolo de controle de *handoff* de uma rede sem fio baseada na tecnologia ATM (WATM). Nesse protocolo, as estações base controlam o roteamento à medida que as unidades móveis trocam de células. Nesse exemplo, pretende-se analisar a metodologia quando aplicada ao projeto de um protocolo dominado por fluxo de controle.

3.1. O protocolo

A expansão das telecomunicações, principalmente no uso de celulares, vem reforçando o interesse por redes sem fio. Novas e complexas aplicações para este tipo de redes aparecem a cada instante juntamente com novos tipos de dispositivos. Computadores móveis ocupam cada vez mais espaço no mercado internacional e existe uma necessidade crescente por taxas de transmissão elevadas.

Em redes móveis, um problema comumente encontrado é a passagem de um terminal móvel de uma célula para outra durante a conversação ou troca de mensagens. Esse deslocamento entre as células denomina-se *handoff* e pode resultar numa simples perda de pacotes ou na perda da conexão.

Nesta seção o desempenho do protocolo é analisado utilizando-se a seqüência de primitivas fornecidas em Acharya [15]. Nesse tipo de protocolo, a localização de um terminal em relação à rede não pode ser determinada apenas através de seu endereço. Esquemas adicionais de endereçamento são necessários para localizar e rastrear os terminais móveis, juntamente com modificações apropriadas no processo de iniciação da conexão. Um protocolo de controle de *handoff* eficiente estabelece novas rotas dinamicamente, ao invés de estabelecer toda a conexão novamente.

3.2. Especificação do Protocolo

A figura 1 apresenta a especificação do protocolo dada por uma rede de Petri condição-ação. Primeiramente, é enviada uma primitiva de *handoff_request (select cos)* para a estação base a qual o terminal móvel está conectado. Esta estação base envia uma primitiva *handoff_request (setup)* para uma outra estação base que poderá acolher este terminal móvel. Neste ponto, as duas estações trocam algumas primitivas e a estação base inicial avisa ao terminal móvel (*handoff_response*) da possível troca de célula.

O terminal envia um *handoff_confirm* para a estação inicial que responde com um *handoff_confirm_complet* confirmando a nova configuração. Por fim, o terminal envia um *handoff_join* à nova estação base que responde com a primitiva *handoff_join_complet* estabelecendo a nova conexão.

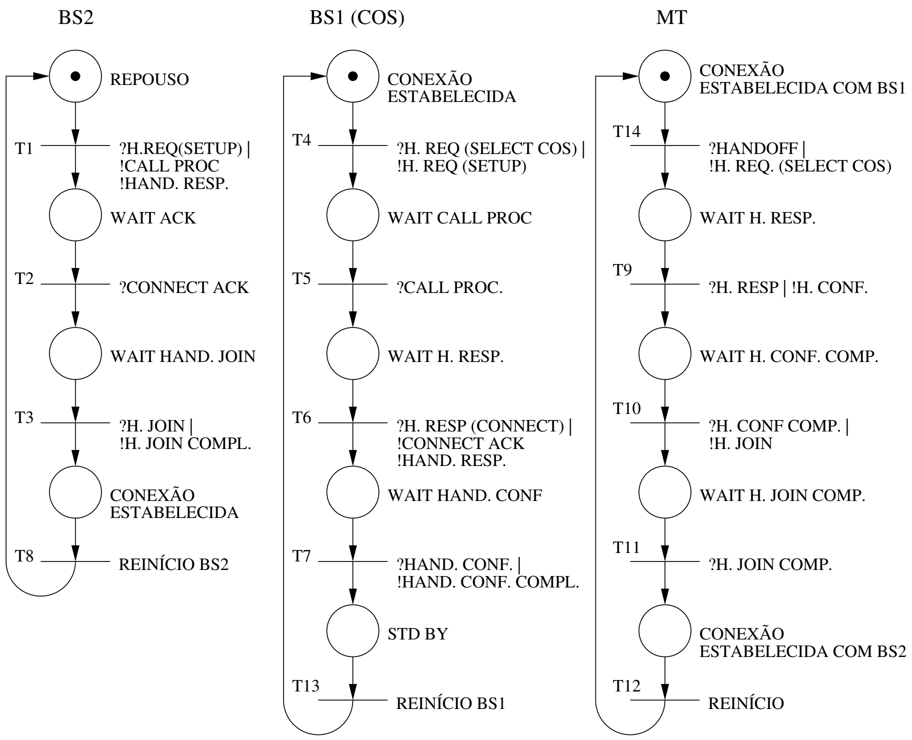

Figura 1: Rede de Petri condição-ação do Protocolo de controle de *Handoff*.

3.3. Avaliação do Desempenho e Escolha da Partição

A medida de desempenho utilizada é a probabilidade de perda de pacotes em função da taxa de transmissão. O atraso ocasionado pelo *handoff* (T_d) fornece o tempo mínimo para se alterar a tabela de roteamento para múltiplos VCs. Esse tempo é diretamente proporcional à taxa de perda causada pelo procedimento de *handoff*. No modelo utilizado em Yuan [16], uma perda ocorre quando uma mudança na tabela de roteamento é realizada durante a transmissão de um fluxo de células dentro do mesmo pacote. Considere que o evento de *handoff* ocorre aleatoriamente dentro do intervalo $T = 1/R$, onde T é o tempo de transmissão de um pacote e R é a taxa de transmissão de pacotes. Portanto, a probabilidade de perda de pacotes P é dada por $P = T_d * R$. A figura 2 mostra o gráfico P versus R.

Devido à simplicidade da curva, 100 indivíduos foram suficientes para a con-

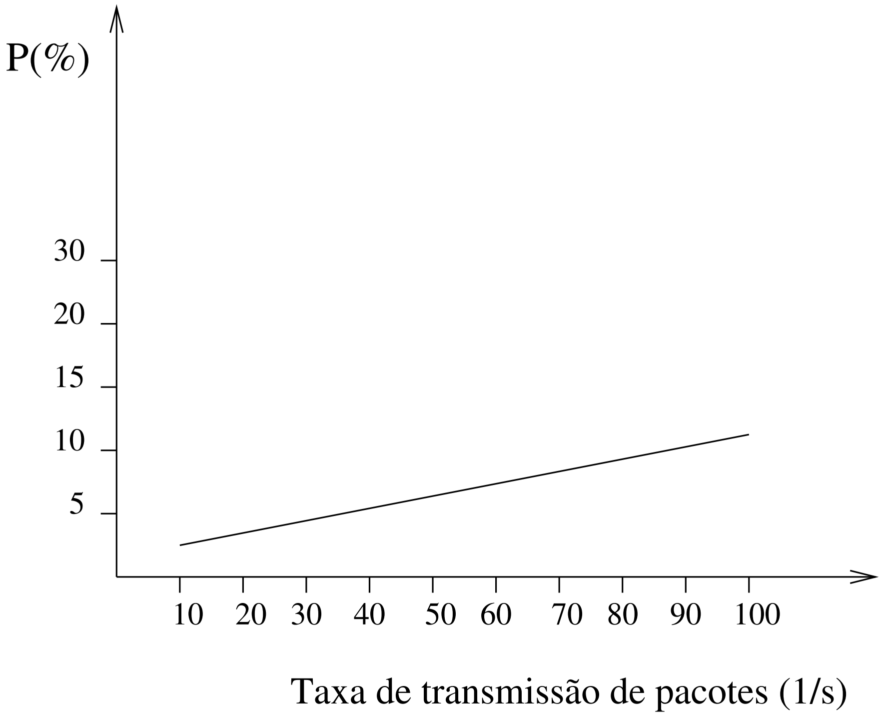

Figura 2: Probabilidade de Perda em função da taxa de transmissão.

vergência do algoritmo. O número de gerações utilizado como critério de terminação é igual a 10. Esses valores foram determinados após vários testes com diferentes tamanhos de populações e gerações. A função objetivo é a diferença entre a probabilidade de perda obtida e a probabilidade de perda desejada, ou seja, os parâmetros do protocolo são extraídos aproximando-se o gráfico da figura 2. Os resultados são apresentados na tabela 1. O erro encontrado para a soma das diferenças entre as probabilidades desejadas e obtidas foi de 0,006.

Utilizando as taxas da tabela 1 e as probabilidades em estado estacionário calculadas pelo Tangram-II, o mesmo critério de partição da seção 2.4 pode ser aplicado e o produto $\lambda_i * \pi_j$ calculado para cada um dos eventos (transições) do protocolo. Os resultados são mostrados na tabela 2.

As tarefas do protocolo relacionadas às transições t_4, t_5, t_6, t_7, t_{12} e t_{13} devem ser implementadas em HW porque as respectivas transições possuem um maior produto $\lambda_i * \pi_j$, enquanto aquelas relacionadas às demais transições, com um produto menor,

Transição	**Taxa (1/s)**
t_1 (λ_1)	42,34
t_2 (λ_2)	60,12
t_3 (λ_3)	421,63
t_4 (λ_4)	344,91
t_5 (λ_5)	477,38
t_6 (λ_6)	397,72
t_7 (λ_7)	14,31
t_8 (λ_8)	156,63
t_9 (λ_9)	20,76
t_{10} (λ_{10})	8,43
t_{11} (λ_{11})	114,90
t_{12} (λ_{12})	338,55
t_{13} (λ_{13})	474,19
t_{14} (λ_{14})	68,03

Tabela 1: Taxas da rede de Petri do protocolo de *Handoff* obtidas pelo AG.

podem ser implementadas em SW. Essa é a primeira partição a ser avaliada em relação ao custo de implementação em HW.

De acordo com os resultados obtidos e analisando-se o modelo da figura 1, as transições a serem implementadas em HW aceleram o processo de *handoff* porque são aquelas associadas às tarefas de mudança da tabela de roteamento na *cross-over switch* e ao restabelecimento da conexão do terminal móvel com a nova estação base. Isso indica que o comportamento do modelo aproxima o mecanismo real de protocolos de controle de *handoff* encontrados em diversas implementações de redes sem fio.

Evento	Produto
$\lambda_1 * \pi_1$	1.52
$\lambda_2 * \pi_2$	3.05
$\lambda_3 * \pi_3$	5.22
$\lambda_4 * \pi_4$	97.61
$\lambda_5 * \pi_5$	14.31
$\lambda_6 * \pi_6$	27.75
$\lambda_7 * \pi_7$	70.32
$\lambda_8 * \pi_8$	0.20
$\lambda_9 * \pi_9$	1.17
$\lambda_{10} * \pi_{10}$	0.17
$\lambda_{11} * \pi_{11}$	1.88
$\lambda_{12} * \pi_{12}$	80.21
$\lambda_{13} * \pi_{13}$	13.60
$\lambda_{14} * \pi_{14}$	3.67

Tabela 2: Valores obtidos para o produto $\lambda_i * \pi_j$.

3.4. Avaliação da Partição

As descrições VHDL de cada transição do protocolo são usadas como entrada para as ferramentas Synopsys e ALTERA, ou seja, tanto para a implementação em *standard cells* quanto em PLDs. No primeiro caso, a síntese foi realizada a partir de uma biblioteca padrão ES2 com tecnologia 0.7μm. A implementação em ALTERA utilizou o compilador de projeto Max+PlusII. A tabela 3 mostra as medidas de área e atraso obtidas para cada uma das transições de estado do protocolo.

A partição proposta é avaliada tomando os valores apresentados e considerando uma área especificada para o custo de, por exemplo, 1 mm^2. Dessa forma, pode-se notar

Transição	Synopsys Área (mm^2)	Synopsys Atraso	ALTERA Atraso
1	0.150	20 ns	60 ns
2	0.015	5 ns	30 ns
3	0.023	5 ns	30 ns
4	0.223	16 ns	60 ns
5	0.015	5 ns	30 ns
6	0.130	20 ns	60 ns
7	0.023	5 ns	30 ns
8	0.015	5 ns	30 ns
9	0.024	5 ns	30 ns
10	0.015	5 ns	30 ns
11	0.015	5 ns	30 ns
12	0.015	5 ns	30 ns
13	0.015	5 ns	30 ns
14	0.223	16 ns	60 ns

Tabela 3: Medidas de área e atraso.

que todas as transições indicadas pelo processo de partição (t_4, t_5, t_6, t_7, t_{12} e t_{13}) podem ser implementadas em HW utilizando-se *standard cells* ou PLDs pois todas as medidas de área são menores do que 1 mm^2 e os atrasos associados satisfazem as taxas calculadas pelo AG.

3.5. Análise do Desempenho da Partição Escolhida

A figura 3 compara o desempenho do protocolo de *handoff* totalmente implementado em *software*, em *standard cells* e em PLDs (ALTERA) com o desempenho da partição HW/SW escolhida. Para isso, duas implementações da partição escolhida foram realizadas: a primeira, usando parte das transições implementadas em PLDs (aquelas designadas para *hardware*) e parte implementada em *software*. A segunda, usando parte das transições implementadas em *standard cells* e parte em *software*. O parâmetro de desempenho utilizado para comparar essas implementações é a probabilidade de perda em função da taxa de transmissão.

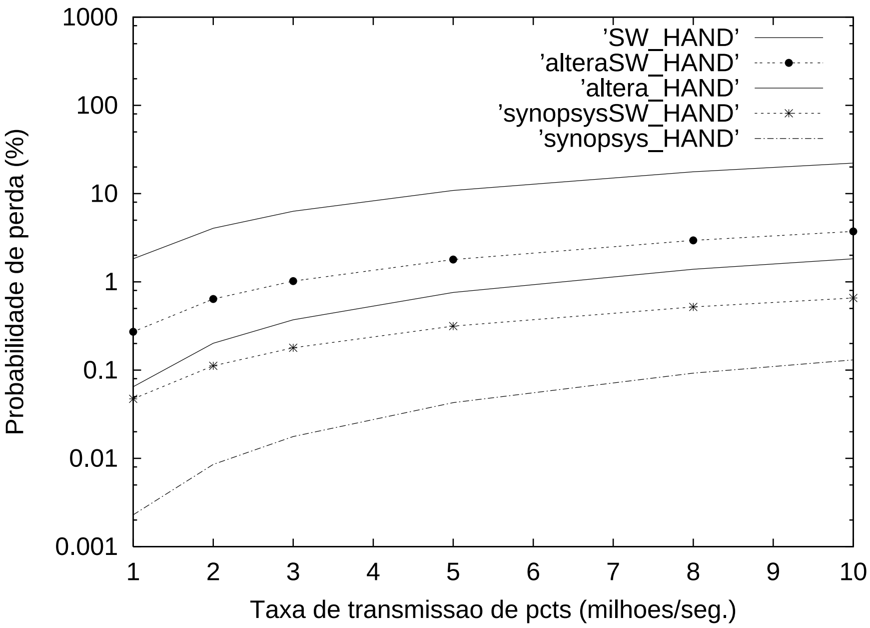

Figura 3: Comparação entre as diversas implementações.

Pode-se observar que a probabilidade de perda na implementação utilizando *standard cells* é aproximadamente 900 vezes menor do que a probabilidade de perda encontrada para a implementação em *software*. Porém, o custo deste tipo de tecnologia é alto, devido à necessidade de se produzir um *layout* do circuito e enviá-lo a uma fábrica de circuitos para a produção dos *chips*. A implementação em PLDs fornece perdas maiores que a implementação em *standard cells*, porém menores do que as taxas encontradas na implementação em SW. Não obstante, implementações em PLDs são ferramentas importantes para prototipagem rápida de sistemas de comunicação a um custo baixo. Sistemas são criados em um *chip* reprogramável adquirindo uma placa PLD que pode ser acoplada ao computador. Possíveis modificações em seu sistema são efetuadas rapidamente através de programação. Ao contrário, modificações no projeto em *standard cells* necessitam de um novo ciclo de fabricação.

A partição HW/SW utilizando PLDs fornece resultados melhores do que a implementação do protocolo todo em *software*. A partição HW/SW utilizando *standard cells* fornece resultados muito melhores do que a implementação do protocolo todo em *software* e melhores do que a partição HW/SW com PLDs. Porém, neste caso, a solução utilizando PLDs satisfaz os requisitos de desempenho com um custo menor do que a implementação em *standard cells*. Caso seja necessário obter um desempenho ainda melhor em detrimento do custo do projeto, pode-se optar pela partição HW/SW utilizando *standard cells*.

4. Considerações Finais

O crescente tráfego das redes atuais causado por aplicações multimídia e *multicast*, entre outras, gerou uma demanda por redes de alta velocidade e por protocolos com altas vazões. Por outro lado, verifica-se uma tendência, nos próximos anos, de uma taxa de crescimento bem maior da largura de faixa dos enlaces do que das velocidades de processamento nos nós dessas redes. Isso gera a necessidade de se investigar soluções que utilizam *hardware* na implementação dos protocolos executados nesses nós. Nesse contexto, o processo de *HW/SW codesign* têm grande importância, permitindo a detecção de erros nas etapas iniciais do projeto. No entanto, esse processo introduz um problema de alta complexidade: decidir quais tarefas do protocolo devem ser implementadas em HW e quais devem ser implementadas em SW, ou seja, definir a partição HW/SW.

A necessidade da constante interferência do projetista no processo de escolha da melhor partição HW/SW, utilizando apenas a sua experiência, torna essa escolha muito subjetiva e o processo de implementação muito lento e sujeito a erros. É importante que o projetista disponha de medidas e critérios objetivos que o auxiliem a tomar uma decisão objetiva e precisa. Com essa finalidade, a contribuição principal deste trabalho é o desenvolvimento de uma metodologia que auxilia o projetista no processo de escolha da partição HW/SW. Essa metodologia baseia-se nos seguintes conceitos e ferramentas: *HW/SW codesign*, redes de Petri estocásticas, na ferramenta de análise de desempenho Tangram-II, nos algoritmos genéticos e em duas ferramentas de síntese de HW, o Synopsys e o Altera. A integração dessas ferramentas em um único processo é um importante resultado desse trabalho. Outro fator importante que diferencia a metodologia apresentada das metodologias encontradas na bibliografia é a utilização de medidas de desempenho no processo de otimização realizado pelo algoritmo genético **desde o início do ciclo de projeto**. A metodologia também estabelece um critério que relaciona uma saída do AG com uma determinada partição HW/SW. Esse critério se mostra coerente ao longo dos diversos exemplos estudados, conforme poderá ser verificado posteriormente.

Embora o AG tenha um alto custo computacional, ele é simples de ser implementado e evita funções de erro mais complexas como, por exemplo, a função exponencial utilizada no método de otimização *Simulated Annealing* [17].

Protocolos que possuem altos custos computacionais com seu desempenho diretamente relacionado à velocidade de processamento nos nós são os melhores candidatos a utilizarem a metodologia de projeto apresentada neste trabalho. Este é o caso do protocolo de controle de *handoff* para WATM, utilizado como estudo de caso. Os resultados obtidos mostram que a partição HW/SW escolhida aumenta o desempenho do protocolo comparado à sua implementação em SW e o custo desta implementação HW/SW é menor do que sua implementação inteiramente em HW.

O custo da implementação em HW pode ser significativamente reduzido caso sejam utilizados dispositivos lógicos programáveis (PLDs), que aproximam o custo do HW do custo do SW, pois o *hardware* pode ser desenvolvido e testado em um programa comercial desenvolvido com essa finalidade (Altera) e sintetizado em pastilhas de baixo custo, ao contrário da implementação em *standard cells*.

Referências

[1] Extremenetworks, "Technology - Hardware Architecture", *Internet Draft*, 2000. `http://www.extremenetworks.com`.

[2] D. Gajski and F. Vahid, "Specification and Design of Embedded Software/Hardware Systems", *IEEE Design and Tests of Computer*, vol. 12, no. 1, pp. 53–67, Jan. 1995.

[3] "Overview of the Ptolemy Project - Computer Science Department - University of California". `http://ptolemy.eecs.berkeley.edu`.

[4] J. Madsen, J. Grode, P. V. Knudsen, M. E. Petersen and A. Haxthausen, "LYCOS: the Lyngby Co-Synthesis System", *Design Automation for Embedded Systems*, vol. 2, no. 2, pp. 1–43, Feb. 1997.

[5] P. Maciel, E. Barros and W. Rosenstiel, "Estimating Functional Unit Number in the PISH Codesign System by Using Petri Nets", in *XII Symposium on Integrated Circuits and Systems Design*, pp. 32–35, Sept. 1999.

[6] M. N. de Miranda, *Uma Metodologia de HW/SW Codesign de Protocolos de Comunicação Baseada na Otimização de Desempenho por Algoritmos Genéticos*. PhD thesis, Programa de Engenharia Elétrica - COPPE/UFRJ, Aug. 2002.

[7] A. P. C. Silva, "Tangram-II User's Manual", tech. rep., Universidade Federal do Rio de Janeiro, Oct. 2000. `http://www.land.ufrj.br`.

[8] IEEE Std 1076-1987, *IEEE Standard VHDL Language Reference Manual*, Mar. 1988.

[9] Synopsys, Inc., *Synopsys Online Documentation, v1998.02*, 1998.

[10] Altera, Corp., *Data Book and Max + PlusII Getting Started*, 1997.

[11] H. Choi, V. Kulkarni and K. Trivedi, "Markov regenerative stochastic Petri nets", *Performance Evaluation*, vol. 20, no. 1, pp. 337–357, May 1994.

[12] "ARP - Analisador/Simulador de Redes de Petri - LCMI - Departamento de Engenharia Elétrica da Universidade Federal de Santa Catarina".

http://www.ppgia.pucpr.br/ maziero/petri/manual-pt.html.

[13] M. Mitchell, *An Introduction to Genetic Algorithms*. MIT Press, 1996.

[14] J. I. Hidalgo and J. Lanchares, “Functional Partitioning for Hardware/Software Codesign using Genetic Algorithm”, in *Proceedings of the 23rd Euromicro Conference*, 1997.

[15] A. Acharya, J. Li and A. Bakre, “Design and Prototyping of Location Management and Handoff Protocols for Wireless ATM networks”, in *ICUPC’97*, 1997.

[16] R. Yuan, K. Biswas, L. J. French, J. Li and D. Raychaudhuri, “A Signaling and Control Architecture for Mobility Support in Wireless ATM Networks”, *MONET*, vol. 1, no. 3, pp. 287–298, Mar. 1996.

[17] A. Laarhoven, *Simulated Annealing: theory and applications*. D. Reidel, 1987.