# TOPSEC – UM PROCESSADOR COM SUPORTE A MUDANÇA DE TOPOLOGIA E PRIMITIVAS DE SEGURANÇA

Bruno Moreira Cestari, Edward D. M. Ordonez
USP – Universidade de São Paulo
UNIVEM – Centro Universitário Eurípides de Marília
UEA – Benq, Manaus, Brazil
brunocestari@terra.com.br; edwdavid@gmail.com

## Resumo

Este artigo apresenta o projeto do ToPSeC, um processador específico para envio e recebimento de pacotes próprio, que suporta a mudança de topologia, dentre elas Anel Unidirecional e Árvore Binária, inclusive com a busca por um caminho alternativo dentro da topologia escolhida e a inclusão de primitivas de segurança. Dessa forma, este projeto mostra o desenvolvimento de um processador que permite uma comunicação eficiente nesse cenário e mostra o impacto causado pela inclusão de segurança nesse contexto.

## 1. Introdução

Com a crescente utilização das redes de computadores em diversos segmentos da sociedade, a transmissão dos dados e a segurança da informação tornam-se pontos importantes, impulsionando tendências e pesquisas de soluções nesse segmento [10,11].

O grande gargalo das comunicações têm sido os equipamentos de rede, como roteadores, *hubs* e *switches*, etc. Esse tipo de dispositivo centraliza o tráfego de pacotes e muitas vezes são os responsáveis pela falta de eficiência da rede [6].

Neste sentido, surge o desenvolvimento de processadores de rede, que são dispositivos que são capazes de realizar o processamento de pacotes de maneira mais rápida e eficiente [11].

Esses dispositivos possuem como características o processamento em tempo real, segurança, *store and forward*, manipulação de pacotes, flexibilidade e escalabilidade, permitir troca de pacotes de tamanhos diferentes em redes com diferentes protocolos e topologias e por fim, capacidade de aprendizado [10].

O segmento de processadores de rede possui várias pesquisas e investimentos, inclusive de soluções proprietárias, principalmente pelos ganhos obtidos e pela tendência de ganhos nesse mercado, que já gira em torno de milhões de dólares [1].

Atualmente, as implementações de processadores de rede existentes são baseadas nas seguintes tecnologias:

- ASIC *(Application Specific Integrated Circuit);*
- Co-processador;
- GPP (*General Purpose Processor*);
- FPGA (*Field Programmable Gate Array*);

Dentre as tecnologias apresentadas acima, fazendo uma comparação de flexibilidade x desempenho, tem-se o seguinte quadro representado na figura 1 [10].

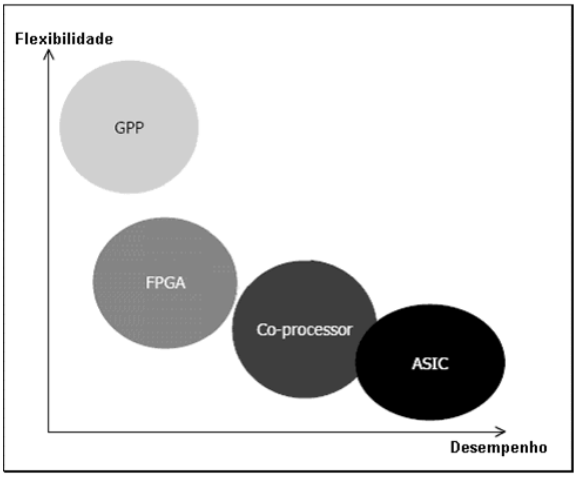


**Figura 1 – Comparação flexibilidade x desempenho**

Baseado nas características das tecnologias apresentadas e fazendo uma comparação de flexibilidade x desempenho, a tecnologia escolhida para este projeto foi a FPGA [4]. Além de oferecer uma fácil prototipação, permitindo a sua reprogramação a cada alteração de projeto, influenciando inclusive no preço, já que é uma tecnologia barata e não oferece custo para a manutenção do projeto, oferece a possibilidade de diversas implementações, que no caso deste projeto foram feitas em várias versões.

Apesar dos processadores de rede agilizar o funcionamento de uma rede, tornando-a mais eficiente, poucos desses dispositivos oferecem suporte à segurança da informação que é manipulada, principalmente pelo fato do aumento da utilização das redes de comunicação em diversos segmentos da sociedade, transmitindo muitas vezes informações pessoais que necessitam permanecerem seguras [5].

Visando a transmissão segura de informação em uma rede de computador, motivado pela baixa gama de opções de processadores de rede que oferecem essa funcionalidade, foi escolhido dois algoritmos de criptografia simétrico para este projeto (DES e AES),

pois há a preocupação com a velocidade na transmissão da informação, para acrescentar segurança através da criptografia e medir o impacto que a inclusão que um algoritmo criptográfico causa a um sistema.

## 2. Trabalhos correlatos

O processador ToPSeC foi desenvolvido baseado no NPSoC – Um novo processador de rede [9] e no Criptoprocessador VLIW [8].O NPSoC é um processador de rede implementado em VHDL e prototipado em FPGA. Baseado no RCNP – Processador de Rede com Suporte a Multi-protocolo e Topologias Dinâmicas [2] e no R2NP (*Reconfigurable RISC Network Processor*) – Processador de Rede RISC Reconfigurável [3]. O NPSoC possui as características do RCNP, mas com modificação da proposta inicial devido à complexidade na implementação das *microengines* implementadas no R2NP, dificultando assim a visualização de seu funcionamento. Seu conjunto de instruções segue o modelo RISC, onde, com apenas 32 instruções foi possível descrever alguns algoritmos de roteamento presentes nos NPs comerciais atuais [9].

O criptoprocessador foi projetado para executar preferencialmente algoritmos de criptografia simétricos e para isso, módulos especiais foram descritos e projetados de maneira otimizada, visando aumentar o desempenho e simplificar a implementação. O Criptoprocessador foi descrito utilizando a linguagem VHDL, suportando uma série de algoritmos simétricos que utilizam chaves de 128 bits ou maior. É importante salientar que os módulos especiais, que diferenciam este criptoprocessador, não são específicos para um determinado algoritmo de criptografia. Estes foram projetados de forma a serem pré-configurados de acordo com as características do algoritmo que será executado [6].

Paralelamente ao desenvolvimento do ToPSeC e também tendo por base o processador de rede NPSoC e as instruções criptográficas correntes no Criptoprocessador VLIW, há também o desenvolvimento do PERS – Processsador específico para rede de sensores [7].

O PERS é um processador simples com o propósito de coletar informações de um determinado local através de sensores, processá-las e enviá-las para a estação base para assim executar ações necessárias de acordo com as informações coletadas. Diferentemente do NPSoC, para se adequar a uma rede de sensores, utiliza apenas a topologia ponto-a-ponto [7].

## 3. ToPSeC – Um processador com suporte a mudança de topologia

Com base nos trabalhos correlatos estudados, foi definido um conjunto reduzido e específicos de instruções que dá suporte ao funcionamento do processador através de seus registradores, ULA (Unidade Lógica Aritmética), a UC (Unidade de Controle), PC (Contador de Programa), RI (Registrador de Instruções) e as instruções específicas para o roteamento da informação na topologia escolhida de maneira eficiente e segura.

O ToPSeC é um processador específico para o envio e recebimento de pacotes de informação próprio nas topologias Anel Unidirecional e Árvore Binária, permitindo a escolha dentre uma dessas topologias e a busca por um caminho alternativo no caso de um nó não estiver disponível. Também oferece segurança às informações transmitidas através da criptografia através dos algoritmos criptográficos simétricos DES ou AES.

Este processador foi implementado em várias versões, sempre adicionando uma nova funcionalidade à versão anterior. Na versão inicial foi realizada a implementação da topologia Anel Unidirecional. A segunda versão foi implementada da topologia Árvore Binária. Na terceira versão foi a implementação da escolha dentre essas duas topologias. Na quarta versão foi a implementação da busca por um caminho alternativo na topologia escolhida. E finalmente na quinta e última versão foi realizada a implementação dos algoritmos de criptografia DES e AES.

Sua arquitetura foi modificada em relação ao NPSoC de modo que se adequasse as necessidades específicas de funcionamento do ToPSeC, principalmente com relação à topologia e à segurança.

Sua arquitetura ficou definida da seguinte forma:

- 1 porta de entrada, nomeada de E1 de 76 bits (DES) e 140 bits (AES);
- 3 portas de entrada, nomeadas de E2 a E4 de 16 bits;
- 1 porta de saída, nomeada de S1 de 76 bits (DES) e 140 bits (AES);
- 3 portas de saída, nomeadas de S2 a S4 de 16 bits;
- 1 registrador de propósito geral de 64 bits (DES) e 128 bits (AES) - (Reg. A);
- 3 registradores de propósito geral de 8 bits (Reg. B, C e D);
- 2 registradores específicos para roteamento de 4 bits (Reg. ID e TIPO);
- 3 registradores específicos para roteamento de 8 bits (Reg. OD, CMP e POSIC);
- Unidade Lógica e Aritmética (ULA);
- Contador de Programa (PC);
- Unidade de Controle (UC);
- Registrador de Instruções (RI).

Na figura 2 verifica-se que a Unidade de Controle controla todo o processador. Neste processador, usou-se o método de FSM (*Finite State Machine*), onde seus estados são bem definidos e respeitam o ciclo de busca, decodificação e execução. O ciclo básico se executa da seguinte maneira. Inicialmente é feita a busca da primeira instrução na memória e soma-se 1 ao PC.

Depois é feita decodificação da instrução e posterior execução pela ULA. Logo em seguida é dado seqüência nas seguintes instruções, até o fim do código *assembly*.

Além da arquitetura, também é importante salientar as informações que esse processador envia e recebe.

**Tabela 1 – Conjunto de instruções do ToPSeC**

**Lógicas e Aritméticas**

| | | | |
|---|---|---|---|
| **AND** | Realiza a operação E entre três registros | **SUB** | Subtrai três registros |
| **OR** | Realiza a operação OU entre três registros | **INC** | Incrementa um registro em +1 |
| **XOR** | Idem OR, porém nega o resultado | **DEC** | Decrementa um registro em -1 |
| **ADD** | Soma três registros | **NEG** | Nega um registro |

**Movimentação**

| | | | |
|---|---|---|---|
| **ENT** | Armazena a origem, o destino e o dado | **LID** | Carrega o endereço e a posição do nó atual |
| **TOPO** | Informa a topologia a ser utilizada | **INF** | Armazena o nó atual, o destino e o dado |

**Desvio**

| | | | |
|---|---|---|---|
| **JUMP** | Desvio incondicional | **JDEST** | Desvia se o dado é para o nó atual |
| **JMPDSI** | Desvia se nível do destino for menor ou igual ao nó atual | **JMPDIA** | Desvia se o lado do destino for igual ao nó atual |
| **JMPDB** | Desvia se nível do destino for maior que o nó atual | **JMPDDA** | Desvia se o lado do destino for diferente do nó atual |
| **JMPN** | Desvia para a instrução SEND com o maior vizinho correto | **JMPANUN** | Desvia se a topologia informada for Anel Unidirecional |
| **JMPARBN** | Desvia se a topologia informada for Árvore Binária | **JMPNXT** | Desvia se o próximo nó estiver pronto para receber a informação |

**Comparação**

| | | | |
|---|---|---|---|
| **CMPSN** | Compara o menor vizinho | **CMPBN** | Compara os maiores vizinhos |
| **CMPLN** | Compara o vizinho anterior | **CMPALT** | Compara o vizinho alternativo caso o próximo não puder receber a informação |

**Envio**

| | | | |
|---|---|---|---|
| **SEND** | Envia a informação através da porta de saída ao nó verificado | **SNDNXT** | Envia requisição ao próximo nó |
| **AWRBFR** | Responde a requisição enviada | | |

**Criptografia**

| | | | |
|---|---|---|---|
| **CRIPTA** | Cifra o dado | **DECRIPTA** | Decifra o dado |

Para essa finalidade, este processador possui quatro portas de entrada, nomeadas de E1 a E4 e quatro portas de saída, nomeadas de S1 a S4, por onde o ToPSeC envia e recebe pacotes de dados específicos para determinada finalidade.

As portas E1 e S1, com tamanhos que variam de 76 bits para a implementação do DES a 140 bits para a implementação do AES. Essas duas portas são responsáveis pelo envio e recebimento da informação que é enviada de nó a nó até o seu destino.

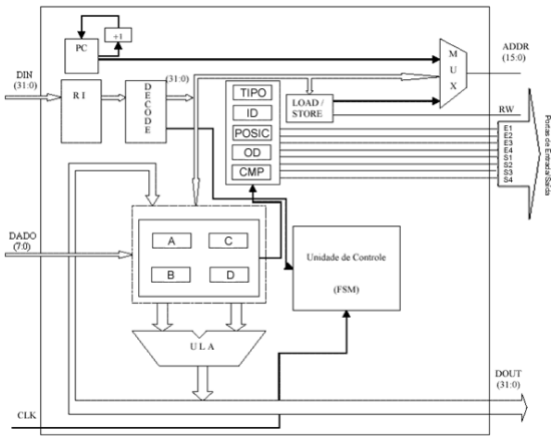


**Figura 2 – Arquitetura ToPSeC**

A mensagem de requisição, que é enviada e recebida pelas portas de saída E2 e S2 é referente à requisição que é feita na situação em que um nó verifica se o seu próximo receptor está apto a receber a informação.

Finalizando, a mensagem referente ao novo mapeamento que os nós envolvidos vão receber para alterar nos seus mapeamentos, que é enviada e recebida pelas portas E3 e S3 e o complemento pelas portas E4 e S4.

Um ponto importante a ressaltar em relação à arquitetura do ToPSeC é com relação ao número de nós utilizados.

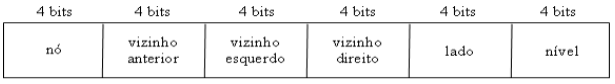


**Figura 3 – Formato da memória ROM**

Por questão de padronização foi definido que as duas topologias teriam um total de 15 nós, de modo que o endereçamento seria de 4 *bits*, inclusive padronizando o tamanho dos módulos de memória ROM e RAM que contém os mapeamentos das topologias utilizadas neste projeto, conforme figura 3.

O conjunto de instruções do ToPSeC segue o modelo RISC, onde ao todo são 31 instruções, que foram desenvolvidas para efetuar transmissão de dados específicos nas diversas implementações das versões constantes nesse processador, que são as implementações das topologias Anel Unidirecional e Árvore Binária, a escolha dentre elas, a busca por um caminho alternativo em uma dessas topologias no caso da indisponibilidade de algum nó e a criptografia da

informação utilizando os algoritmos DES e AES.

## 4. Resultados – Simulação e estatísticas de prototipação e desempenho

Neste projeto, com a preocupação do desenvolvimento e implementação das funcionalidades de maneira cumulativa, onde a cada nova implementação de uma nova versão, as funcionalidades implementadas anteriormente também poderiam ser utilizadas, de modo que no último programa *assembly* é possível a execução de qualquer implementação, conforme código de recebimento do dado tabela 2.

Comparando os resultados da versão 5, que contém as implementações de criptografia com a primeira versão desenvolvida no início da pesquisa, contendo apenas a implementação da topologia Anel Unidirecional, é possível verificar um aumento significativo de recursos, principalmente de CLBs, devido à inclusão de criptografia, mostrando o impacto da criptografia nesse projeto.

A medida temporal do processador, quanto ao tempo de propagação no circuito é de 24,798 ns o que corresponde a uma freqüência máxima do processador de 40,035 MHZ para o DES e 28,652 ns para uma freqüência máxima de 34,902 Mhz para o AES, enquanto a quarta versão, sem criptografia com todas as outras implementação foi de 17,536 ns para uma freqüência máxima de 57,026 Mhz.

**Tabela 2 – Programas de envio e recebimento da informação**

| | Recebimento do dado | Ciclos | | Envio do dado | Ciclos |
|---|---|---|---|---|---|
| 0 | TOPO 1/2 | 2 | 0 | TOPO 1/2 | 2 |
| 1 | LID 2,12 | 2 | 1 | INF 5AAA | 2 |
| 2 | AWRBFR | 2 | 2 | CRIPTA | 16 / 14 |
| 3 | ENT | 2 | 3 | AWRBFR | 2 |
| 4 | JDEST 24 | 2 | 4 | JMPARBN 7 | 2 |
| 5 | JMPARBN 8 | 2 | 5 | CMPLN | 3 |
| 6 | CMPLN | 3 | 6 | JMPANUN 17 | 2 |
| 7 | JMPANUN 18 | 2 | 7 | JMPDSI 9 | 3 |
| 8 | JMPDSI 10 | 3 | 8 | JMPDB 10 | 3 |
| 9 | JMPDB 11 | 3 | 9 | CMPSN | 3 |
| 10 | CMPSN | 3 | 10 | JMPN 17 | 2 |
| 11 | JMPN 18 | 2 | 11 | JMPDIA 13 | 3 |
| 12 | JMPDIA 14 | 3 | 12 | JMPDDA 15 | 3 |
| 13 | JMPDDA 16 | 3 | 13 | CMPBN | 3 |
| 14 | CMPBN | 3 | 14 | JMPN 17 | 2 |
| 15 | JMPN 18 | 2 | 15 | CMPBN | 3 |
| 16 | CMPBN | 3 | 16 | JMPN 17 | 2 |
| 17 | JMPN 18 | 2 | 17 | SNDNXT | 2 |
| 18 | SNDNXT | 2 | 18 | JMPNXT 21 | 2 |
| 19 | JMPNXT 22 | 2 | 19 | CMPALT | 3 |
| 20 | CMPALT | 3 | 20 | JUMP 17 | 2 |
| 21 | JUMP 18 | 2 | 21 | SEND | 2 |
| 22 | SEND | 2 | 22 | HALT | 2 |
| 23 | HALT | 2 | | Total | 69 / 67 |
| 24 | DECRIPTA | 16 / 14 | | | |
| 25 | JUMP 23 | 2 | | | |
| | Total | 75 / 73 | | | |

Assim, é possível observar que a inclusão de criptografia influenciou diretamente no desempenho do ToPSeC, pois apresentou uma queda significativa de 29,7% para a implementação como o DES e de 38,7% para a implementação do AES na freqüência do processador em relação a quarta versão. Esta queda é significativa, porém é irrelevante considerando a importância de se inserir segurança na comunicação.

## 5. Conclusões e trabalhos futuros

O grande desafio foi a implementação cumulativa do processador, como se a cada nova versão implementada ele sofresse uma evolução.

Para que isso fosse possível, houve a necessidade de adaptações para que topologias diferentes funcionassem sobre a mesma plataforma, dessa forma possibilitando recursos como escolha de topologia e a busca por caminhos alternativos.

Com relação à comparação dos dois algoritmos de criptografia utilizados neste projeto, fica claro que o DES se mostra mais rápido que o AES em termos de desempenho, porém se for considerado a quantidade de informação cifrada pelo tempo despendido para essa tarefa. O AES se mostra mais vantajoso, pois em 160 ns, correspondente a 10 iterações que realiza, ele cifra 800 Mbits/seg. contra 355 Mbits/seg. de dados cifrados em 180 ns correspondentes a 16 iterações.

Como trabalhos futuros, desejam-se incluir outros algoritmos de criptografia, como RC5 e RSA e outras topologias, como Hipercubo, bem como suporte para o seu funcionamento neste processador, inclusive na escolha de topologia. Também se deseja realizar a implementação para reconfiguração parcial, melhorando a funcionalidade da escolha de topologia e o remapeamento do nó.

## 6. Referências


[1] CWYNAR, T., LIMA, S., "Network Processors and their Impact in Networking Industry". Acessado em agosto de 2000, disponível em "netlab.cs.tsinghua.edu.cn/~ljsheng/ npu/from_servio/np.ppt".

[2] FREITAS, H. C., MARTINS, C. A. P. S., "Processador de Rede com Suporte a Multi-protocolo e Topologias Dinâmicas" – WSCAD 2001.

[3] FREITAS, H. C., MARTINS, C. A. P. S., "R2NP: Processador de Rede RISC Reconfigurável" – WSCAD 2002.

[4] MORENO, E. D., PEREIRA, F. D., PENTEADO, C. G., PERICINI R. A., "Projeto, Desenvolvimento e Aplicações de Sistemas Digitais em Circuitos Programáveis (FPGAs)". Marília: Editora Bless, 2003.

[5] MORENO, E. D., PEREIRA, F. D., CHIARAMONTE, R. B., "Criptografia em Software e Hardware – Implantação e Desempenho". Marília. Editora Novatec. 2005.

[6] MUZZI, Fernando A., "Um Protótipo Modular em FPGA do padrão de Segurança PKCS#11". 2005. Dissertação de Mestrado em Ciência da Computação do PPGCC da UNIVEM, Marília, 2005.

[7] OLIVIRA, Alexande Ponce de, "PERS – Um Processador específico para Redes de Sensores com primitivas de Segurança". Dissertação de Mestrado em Ciência da Computação do PPGCC da UNIVEM, Marília, 2006.

[8] PEREIRA, Fábio D., "Um Criptoprocessador VLIW para Algoritmos Criptográficos Simétricos". 2004. Dissertação de Mestrado em Ciência da Computação do PPGCC da UNIVEM, Marília, 2004.

[9] PRADO, Ricardo P., "NPSoC – Arquitetura e Protótipo de um Novo Processador de Rede". 2004. Dissertação de Mestrado em Ciência da Computação do PPGCC da UNIVEM, Marília, 2004.

[10] SHAH, Niraj, "Understanding Network Processors". Dissertação de Mestrado, University of California, Berkeley, USA, 2001. Disponível em http://www.cs.berkley.edu/~plisher/UnderstandingNPs.pdf. Último acesso: Novembro de 2006.

[11] TANEMBAUM, Andrew S., "Redes de Computadores". Editora Campus, 3º Edição, 1999.

[12] XILINX Development Systems, "Synthesis and Simulation Design Guide – Designing FPGAs with HDL", 1998.